

# LATEX

## 智能化

# 科技排版系统

冯伟国 编著

同济大学出版社

# Latex

**L<sup>A</sup>T<sub>E</sub>X**

# 智能化科技排版系统

冯伟国 编著

同济大学出版社

(沪)新登字 204 号

## 内 容 简 介

L<sup>A</sup>T<sub>E</sub>X 是国际上公认的标准西文排版软件之一,已被许多科技论文杂志和出版社采用。近年来,越来越多的国内外西文杂志社开始接受用户直接在微机上以规范 L<sup>A</sup>T<sub>E</sub>X 语言编排而成的论文文稿。日前,国内也推出了汉化的 L<sup>A</sup>T<sub>E</sub>X 软件(CI<sup>A</sup>T<sub>E</sub>X 系统)。

本书全面而系统地介绍了 L<sup>A</sup>T<sub>E</sub>X 程序设计语言和排版系统的使用方法。书中详细列举了大量的 L<sup>A</sup>T<sub>E</sub>X 排版控制命令的应用程序实例。在附录中,介绍了中西文 CI<sup>A</sup>T<sub>E</sub>X 和国际上最新推出的作图辅助软件 T<sub>E</sub>Xcad 的使用方法。

本书可以作为大专院校本科生和研究生的教材以及中等以上文化程度的读者自学用书,也可作为科技工作者、大专院校教师、涉外文件管理人员和编辑文件的计算机用户的重要参考书。

责任编辑: 莫惠林

封面设计: 李志云

## L<sup>A</sup>T<sub>E</sub>X 智能化科技排版系统

冯伟国 编著

\* \* \*

同济大学出版社出版

(上海市四平路1239号)

新华书店上海发行所发行

同济大学印刷厂印刷

\* \* \*

开本: 787 × 1092 1/16 印张: 17.25 字数: 440 千字

1993 年 9 月第 1 版 1993 年 9 月第 1 次印刷

印数: 1 — 2000 册 定价: 15.00 元

ISBN 7-5608-1297-X/TP·129

# 序

TeX 软件的出名, 是因为它出自计算机科学家 D. K. Knuth 之手, 巧妙的构思, 精心的设计, 已使它成为现代电脑排版软件的先驱和样板, 对其后的研究和发展影响是很大的。LaTeX 系统是 TeX 语言的高层次扩展, 比后者更加智能化和好用, 因此也更便于推广, 是当今国际上最流行的排版软件之一。

TeX 类排版软件特别受到科学工作者的钟爱和青睐, 因为它很适合各类书刊文章的出版需要, 能够排出符合专业要求并且十分复杂的图表和数学公式, 印刷质量精美, 输出版式灵活多变, 几乎可以随心所欲地调整和更改, 把电脑排版的智能特点发挥得淋漓尽致。

由于其非凡的特点, 加上不难掌握, 很快地普及和流行起来, 目前已成为国际公认和普遍接受的标准排版软件之一。许多国际专业学会的期刊和学术会议, 都直接接受用 TeX 系统排版的文稿, 以软盘存贮形式或电子邮件方式传送, 从而缩短了出版周期。现在国内也有不少学者开始直接使用个人计算机编写文稿, 甚至自己动手用 TeX 软件编排清样, 这不仅提高了出稿的速度和质量, 使版面设计更符合作品的内涵要求, 也便于与国际更好地接轨。

应当指出, TeX 本质上是一类排版程序设计语言, 它并不限制使用者的创造性发挥。事实上, 有些科技工作者尽管本人并非程序员或专业排版人员, 却能够通过自己编写 TeX 指令, 领略到如何驾驭电脑进行柔性创作的无穷乐趣。

本书作者冯伟国博士是一位年青有为的物理学家, 他学有专长, 1987 年留学回国后, 一直耕耘在教学和研究的第一线, 是上海市高校的优秀青年教师和霍英东青年教师奖的获得者。他又称得上是一位计算机应用的行家, 无论在理论计算和研究, 或是物理的计算机辅助教学方面, 都取得了出色的成果, 计算机已成为他得心应手的工具。电脑排版并不是他的本行, 只是因为工作需要, 他又以他对新事物的特有敏感性, 为他的工具宝库增添了一个新的帮手, 而且又钻研得加此之深。他不仅自己从中得益了, 还想到让更多的同行也能分享其快乐, 这也许是他作为教师的固有一天性。

冯伟国博士是个有心人, 他做了一件非常有意义和有传值的工作, 广大科技工作者一定会感谢他在推广人类先进科技成果方面的不遗余力的努力。我想, TeX 系统的成功, 正在于设计它的专家最深刻地理解专业人员的需求; 而本书的可贵之处, 就在于作者是一个实际的科技工作者, 他更能从用户的观点和心理, 挖掘 TeX 系统的特点和奥秘。当然, 本书的得益者远不止是那些 LaTeX 系统的未来用户, 即使对那些专业从事排版系统和其它软件系统的开发人员, 也可从中得到不少启发。

中国的计算机应用事业要赶超世界先进水平，还需要通过几代人的共同努力。希望今后能出现更多这样的好书和像冯伟国那样的热心人，以推广计算机应用为己任。作为一名计算机专业工作者，我要向作者很好地学习，踏踏实实地多做实际工作，为广大用户服务，为攀登新的高峰填石铺路。

华东理工大学教授

居德华

1993年8月于上海





## 作者简介

冯伟国原是江西插队的上海知青，77年恢复高考后进复旦深造十年，期间曾作为联合培养的博士生，在美国加州大学旧金山分校留学一年，88年获理学博士学位。现在是同济大学物理系副系主任、副教授、硕士导师。他与合作者在国内外学术刊物上共发表了三十多篇学术论文。作为一维导电聚合物局域模研究工作的合作者之一，88年获国家自然科学奖四等奖；91至93年连年获上海高校优秀青年教师荣誉称号；92年获霍英东青年教师奖(教学类)三等奖。

# 目 录

序	xv
<b>1 系统简介</b>	<b>1</b>
1.1 引言	1
1.2 系统构成	3
1.2.1 L <sup>A</sup> T <sub>E</sub> X 排版系统	3
1.2.2 软件配置	3
1.2.3 西文字库	3
1.3 L <sup>A</sup> T <sub>E</sub> X 系统的主要排版功能	4
1.4 背景知识和必要条件	5
<b>2 L<sup>A</sup>T<sub>E</sub>X 使用入门</b>	<b>7</b>
2.1 基本操作步骤	7
2.1.1 编辑 L <sup>A</sup> T <sub>E</sub> X 源文件	7
2.1.2 执行排版	7
2.1.3 显示校样	8
2.1.4 打印输出	8
2.2 小样文件	8
2.2.1 文件版式命令	9
2.2.2 文稿排版环境起始命令	10
2.2.3 文稿排版环境结束命令	11
2.3 打字与排版入门	11
2.3.1 空格	12
2.3.2 换行	12
2.3.3 分段	12
2.3.4 换页	13
2.3.5 拆字	13
2.3.6 连字	13
2.3.7 标点符号	13

2.3.8 控制字符 .....	15
2.3.9 源程序注释控制符 .....	15
2.4 执行排版 .....	15
2.5 排版输出 .....	18
2.5.1 排版样式的屏幕显示 .....	18
2.5.2 宽行针式打印机输出 .....	20
2.5.3 激光打印机输出 .....	21
<b>3 基本行排命令</b> .....	<b>25</b>
3.1 转码控制符 .....	25
3.1.1 控制字符类命令 .....	25
3.1.2 控制词类命令 .....	26
3.2 字体转换命令 .....	27
3.3 字号转换命令 .....	29
3.4 分组控制符——花括号用途之一 .....	32
3.5 参变量控制符——花括号用途之二 .....	33
3.6 尺寸控制命令 .....	33
3.6.1 L <sup>A</sup> T <sub>E</sub> X 长度计量单位 .....	33
3.6.2 L <sup>A</sup> T <sub>E</sub> X 横向空白间距控制命令 .....	34
3.6.3 L <sup>A</sup> T <sub>E</sub> X 纵向空白间距控制命令 .....	34
3.6.4 长度控制命令的赋值命令 .....	37
3.7 LR 排版盒子 .....	37
3.7.1 无框线LR 排版盒子 .....	38
3.7.2 带框线LR 排版盒子 .....	39
3.7.3 实心矩形盒子 .....	40
3.7.4 LR 排版盒子的版面控制 .....	41
3.8 无框线LR 排版盒子的存取 .....	41
3.8.1 存贮库的命名 .....	41
3.8.2 无框线LR 排版盒子的存贮 .....	42
3.8.3 存贮信息的调用 .....	42
3.9 西文特殊符号的排版控制命令 .....	43
3.10 L <sup>A</sup> T <sub>E</sub> X 约定 .....	44
<b>4 排版环境命令</b> .....	<b>47</b>
4.1 独立排版环境的进入与退出 .....	47
4.2 引文环境命令 .....	48
4.2.1 命令格式1 .....	48
4.2.2 命令格式2 .....	49
4.3 诗句环境命令 .....	51



4.4	行居中环境命令	52
4.4.1	多行居中命令	52
4.4.2	单行居中命令	53
4.5	行右齐和行左齐环境命令	54
4.5.1	行右齐命令	54
4.5.2	行左齐命令	55
4.6	引文环境中的行对齐命令	57
4.7	抄录环境命令	57
4.7.1	多行抄录环境命令	58
4.7.2	单行抄录命令	58
4.8	条目列表环境命令	59
4.8.1	编号条目列表环境	59
4.8.2	无序号条目列表环境	61
4.8.3	描述型编码条目列表环境	62
5	版式处理命令	65
5.1	章节排版命令	65
5.2	脚注命令	67
5.3	旁注命令	69
5.4	页码和书眉排版命令	70
5.4.1	页码数字字体命令	70
5.4.2	页眉版式命令	71
5.5	附录版式命令	72
5.6	列目录命令	73
5.7	题名排版命令	73
5.8	小页排版命令	75
5.8.1	小页排版环境命令	76
5.8.2	小页排版盒子命令	77
5.9	摘要排版控制命令	78
5.10	分栏排版控制命令	78
5.11	版式全局说明控制命令	79
5.11.1	版心尺寸控制命令	79
5.11.2	页边距控制命令	80
5.11.3	页眉尺寸控制命令	81
5.11.4	脚注尺寸控制命令	82
5.11.5	旁注尺寸控制命令	82
5.11.6	行距控制命令	82
5.11.7	段落间距控制命令	83
5.11.8	段首缩排控制命令	83

5.11.9 栏间空距排版命令	84
5.12 版面撑满控制命令	84
5.12.1 横向撑满控制命令	84
5.12.2 纵向撑满控制命令	85
5.12.3 点线撑满控制命令	85
<b>6 数学公式排版命令</b>	<b>87</b>
6.1 数学排版概述	87
6.1.1 行中数学公式状态命令	88
6.1.2 独立数学公式状态命令	89
6.1.3 独立编号数学方程命令	89
6.2 数学公式状态下的基本规则	90
6.2.1 字距	91
6.2.2 字体	92
6.2.3 数学符号	92
6.2.4 公式的分行和行距	92
6.3 上下标命令	92
6.3.1 上标排版命令	92
6.3.2 下标排版命令	93
6.4 数学符号排版命令	94
6.4.1 希腊字母	94
6.4.2 书写体	95
6.4.3 特殊数学符号	95
6.4.4 可变型数学符号	98
6.4.5 数学公式中的省略号	99
6.5 对数型函数排版命令	100
6.6 分式、根式和取模的排版命令	101
6.6.1 分式	101
6.6.2 根式	101
6.6.3 取模	102
6.7 矩阵排版命令	103
6.7.1 矩阵元排版环境命令	103
6.7.2 矩阵的对位参量	104
6.8 界标排版命令	105
6.8.1 静态配置界标	106
6.8.2 动态配置界标	106
6.9 多行数学方程式排版命令	108
6.9.1 多行编号方程式排版命令	108
6.9.2 多行无编号方程式排版命令	109

6.10 数学符号的修饰	110
6.10.1 划横线命令	110
6.10.2 卧式花括号命令	111
6.10.3 “戴帽”命令	112
6.10.4 “堆砌”命令	113
<b>7 表格排版命令</b>	<b>115</b>
7.1 活动表格	115
7.1.1 表格排版环境命令	115
7.1.2 表格标题命令	116
7.1.3 表格目录命令	116
7.2 无框线表格制表命令	117
7.2.1 无框线表格排版环境命令	117
7.2.2 表文排版控制命令	117
7.3 可划线表格制表命令	119
7.3.1 可划线表格排版环境命令	119
7.3.2 表格划线命令	120
7.4 对位表达式	122
7.4.1 q- 表达式	122
7.4.2 p- 表达式	123
7.4.3 *- 表达式	123
<b>8 L<sup>A</sup>T<sub>E</sub>X 绘图命令</b>	<b>125</b>
8.1 插图环境命令	125
8.1.1 插图标题命令	126
8.1.2 插图目录命令	126
8.2 绘图环境命令	126
8.3 基本作图命令	128
8.3.1 “粘贴”文字	128
8.3.2 画矩形	128
8.3.3 画直线	132
8.3.4 画箭头	134
8.3.5 堆积字符串	135
8.3.6 画圆和圆弧	136
8.4 无线矩形图形的存取	139
8.4.1 图形存贮库的命名	139
8.4.2 无线矩形图形的存贮	139
8.4.3 存贮图形的调用	140
8.5 规则平移场合下图形的复制	140

<b>9 交叉引用</b>	<b>143</b>
9.1 章节图表目录	144
9.1.1 章节目录的生成	144
9.1.2 图表目录的生成	144
9.2 章节图表的交叉引用	145
9.2.1 引用标识符	145
9.2.2 章节的交叉引用	146
9.2.3 方程的交叉引用	146
9.2.4 图表的交叉引用	147
9.2.5 页面的交叉引用	147
9.3 参考文献	148
9.3.1 参考文献的编排	148
9.3.2 参考文献的引用	149
9.4 编制索引	150
9.4.1 生成索引信息文件	151
9.4.2 编制文件索引	152
9.5 编制词汇表	153
<b>10 L<sup>A</sup>T<sub>E</sub>X 程序设计高级技巧</b>	<b>155</b>
10.1 自行定义 L <sup>A</sup> T <sub>E</sub> X 控制命令	155
10.1.1 定义一个排版子过程控制命令	155
10.1.2 含变量控制命令的自定义方法	156
10.1.3 重新定义已经设置的控制命令	158
10.2 自行定义 L <sup>A</sup> T <sub>E</sub> X 排版环境	158
10.2.1 排版环境控制命令的化简	158
10.2.2 自行定义排版环境控制命令	159
10.2.3 含变量排版环境的自定义方法	160
10.2.4 重新定义已经设置的排版环境	162
10.3 人机对话	162
10.3.1 终端显示命令	162
10.3.2 终端输入命令	163
10.4 调入另外一个 L <sup>A</sup> T <sub>E</sub> X 源文件	165
10.4.1 无选择方式调入 L <sup>A</sup> T <sub>E</sub> X 子文件	166
10.4.2 有选择调入 L <sup>A</sup> T <sub>E</sub> X 子文件	167
10.5 如何改变系统的计数方式	168
10.5.1 计数器和计数方式	168
10.5.2 为计数器赋值	168
10.5.3 计数器数字打印命令	169
10.5.4 改变序码的字体形式	169

10.6 自定义排版计数器 .....	171
10.7 定理排版环境命令 .....	171
10.8 列表环境(list) 的构造 .....	174
10.9 如何使用新的字体 .....	176
10.10 断行控制命令 .....	179
10.10.1 断字注解命令 .....	179
10.10.2 断行控制命令 .....	179
10.10.3 自然段断字处理环境命令 .....	180
10.11 断页控制命令 .....	181
<b>11 信件类(letter) 版式和 APS 版式</b> .....	<b>183</b>
11.1 信件类版式 .....	183
11.1.1 信件类(letter) 文件版式命令 .....	183
11.1.2 发信人有关通讯信息 .....	184
11.1.3 信件排版环境 .....	185
11.1.4 信件正文的引出和结尾 .....	185
11.1.5 信件附加信息 .....	186
11.1.6 信件排版实例 .....	186
11.2 APS 科技论文版式 .....	187
11.2.1 APS 宏程序库文件的结构和安装 .....	187
11.2.2 APS 文件版式的环境设置 .....	188
11.2.3 APS 版式的尺寸控制 .....	189
11.2.4 题名版面 .....	189
11.2.5 节次标题 .....	192
11.2.6 脚注和参考文献 .....	192
11.2.7 插图与表格版面的设计 .....	193
11.2.8 数学公式的编排 .....	194
<b>12 源程序语法错误的发现和处理</b> .....	<b>197</b>
12.1 出错定位 .....	197
12.2 TeX 出错信息 .....	198
12.3 LaTeX 出错信息 .....	200
12.4 语法严重错误 .....	201
12.4.1 无文稿排版环境 .....	201
12.4.2 无文件版式命令 .....	201
12.4.3 无效文件版式 .....	202
12.4.4 数学状态中的意外语法错误 .....	203
12.5 TeX 系统警告信息 .....	204
12.5.1 断行警告信息 .....	204

12.5.2 断页警告信息 .....	205
12.6 L <sup>A</sup> T <sub>E</sub> X 系统警告信息 .....	205
<b>附录A 英文字体</b>	<b>207</b>
<b>附录B CCT 中西文排版系统简介</b>	<b>209</b>
B.1 CCT 系统的主要功能和特点 .....	210
B.2 CCT 中西文排版系统的构成 .....	211
B.3 C <sup>I</sup> L <sup>A</sup> T <sub>E</sub> X 系统的基本流程 .....	212
B.4 CCT 初始化程序CCTINIT.EXE .....	213
B.5 CCT 预处理程序CCT.EXE .....	214
B.6 字号定义文件CCT.DAT .....	214
B.7 C <sup>I</sup> L <sup>A</sup> T <sub>E</sub> X 专用排版命令 .....	215
B.7.1 C <sup>I</sup> L <sup>A</sup> T <sub>E</sub> X 源文件的版式 .....	215
B.7.2 汉字字体转换命令 .....	216
B.7.3 汉字字号转换命令 .....	216
B.7.4 汉字尺寸控制命令 .....	217
B.7.5 C <sup>I</sup> L <sup>A</sup> T <sub>E</sub> X 与西文L <sup>A</sup> T <sub>E</sub> X 的兼容性 .....	218
B.8 图形接口 .....	219
B.8.1 图形拼版命令 .....	219
B.8.2 BMF 图形文件的生成及数据格式 .....	220
<b>附录C 作图辅助软件 T<sub>E</sub>Xcad 简介</b>	<b>223</b>
C.1 引言 .....	223
C.2 T <sub>E</sub> Xcad 软件运行环境 .....	223
C.3 T <sub>E</sub> Xcad 的启动和退出 .....	224
C.4 菜单选择 .....	225
C.5 信息窗口 .....	226
C.6 使用T <sub>E</sub> Xcad 窗口绘制图形 .....	227
C.6.1 “粘贴”文字 .....	227
C.6.2 实线、虚线和实心矩形盒子 .....	228
C.6.3 直线与箭头 .....	229
C.6.4 圆和圆角矩形 .....	229
C.7 图像编辑 .....	230
C.7.1 图形定义 .....	230
C.7.2 文字编辑 .....	231
C.7.3 图像复制和移动 .....	231
C.7.4 图像删除 .....	231
C.7.5 图形或画面的剪取和拼嵌 .....	231
C.8 其它功能 .....	232

---

C.9 绘图源程序的调入和生成 .....	233
<b>附录D 出错信息</b>	<b>235</b>
D.1 L <sup>A</sup> T <sub>E</sub> X 系统出错信息 .....	235
D.2 T <sub>E</sub> X 系统出错信息 .....	238
<b>附录E 常用 L<sup>A</sup>T<sub>E</sub>X 命令</b>	<b>243</b>
<b>参考文献</b>	<b>247</b>
<b>索引</b>	<b>249</b>
<b>后记</b>	<b>258</b>



# 表格目录

3.1	西文字体转换命令表	27
3.2	字号转换命令表	30
3.3	不同基准字体状态下字号转换命令控制的实际尺寸	31
3.4	各种字号的有效字体详表	32
3.5	常用的 $\text{\LaTeX}$ 长度计量单位表	34
3.6	特殊标识符号命令表	43
3.7	西文(非英语类)特殊符号命令表	44
3.8	字母外形特征符号修饰控制命令表	44
5.1	章节排版命令表	66
5.2	页码形式参数表	71
6.1	数学公式状态字距命令表	91
6.2	希腊字母控制命令表	95
6.3	数学运算符表	96
6.4	确定型逻辑运算符表	96
6.5	箭头符号表	97
6.6	花样符号表	97
6.7	否定式逻辑运算符表	98
6.8	可变型数学符号表	98
6.9	省略号控制命令表	99
6.10	对数型函数排版命令表	100
6.11	矩阵排版环境命令列参数表	103
6.12	矩阵元排版命令对位参数表	105
6.13	常用界标排版命令表	106
6.14	数学符号“戴帽”命令表	112
10.1	$\text{\LaTeX}$ 排版计数器标识名表	168
10.2	序码数字字体控制命令表	170
10.3	常用排版字库文件引用名表	177

---

B.1 汉字字体转换命令表 .....	216
---------------------	-----

# 插图目录

2.1	L <sup>A</sup> T <sub>E</sub> X 排版系统流程图	8
2.2	显示系统的选择菜单	19
2.3	激光打印驱动程序(PTIHP.EXE) 屏幕显示菜单	22
5.1	A4 纸版式全局说明示意图	80
8.1	画实线矩形示意图	130
8.2	虚线矩形作图实例	131
8.3	画直线示意图	133
8.4	矢量(箭头) 绘制示意图	136
8.5	空心圆和实心圆绘制示意图	137
8.6	圆角矩形绘图方式示意图	138
8.7	箭头或直线与圆弧光滑连接示意图	139
8.8	无线矩形图形的存取示意图	140
8.9	箭头线性平移复制示意图	141
8.10	实心圆二维复制示意图	142
10.1	提示信息终端显示示意图	163
10.2	文字信息终端输入示意图	164
10.3	为自定义控制命令赋字符串的屏幕显示示意图	165
10.4	条目列表环境(list) 版面控制示意图	175
C.1	T <sub>E</sub> Xcad 作图窗口示意图	226
C.2	T <sub>E</sub> Xcad 主菜单和子菜单结构图	227

# 1

## 系统简介

### 1.1 引言

现代计算机技术的重要应用之一，是将各类文件、书籍和报刊的编辑排版工作自动化。在过去的十多年中，经过大批杰出的计算机科技工作者的努力，诞生了许多不同风格的排版软件，使得计算机的编辑排版工作越来越完美。在这些排版软件中，由美国著名计算机专家D. E. Knuth 带领的一个研究小组在Stanford 大学经过多年研制成功的 $\text{\TeX}$  科技排版软件独树一帜，自问世以来受到了广大出版商和科技工作者的普遍欢迎，已被许多国际学(协)会及杂志所采用。由于它可以排出非常复杂的数学公式，且版面的规范设计达到了异常精美的程度，如今它已成为国际上公认的标准排版软件之一。

$\text{\TeX}$  软件中设计了900 个排版命令，其中有200 多个是基本命令。要熟练掌握和灵活应用如此多的排版命令，对于非计算机排版专业的用户来说不是一件容易的事。为了简化一些复杂的排版过程，从而使广大用户能方便地使用 $\text{\TeX}$  排版软件，又有人先后利用 $\text{\TeX}$  的排版功能，把 $\text{\TeX}$  的基本命令加以组合，定义了一些新的排版命令(通常称之为宏命令)。这样，在 $\text{\TeX}$  系统基础上开发了许多高级的排版软件系统。在这些高级的软件中，由美国人莱斯利·拉姆波特(Leslie Lamport) 用 $\text{\TeX}$  语言编写成的 $\text{\LaTeX}$  排版系统是最著名的西文排版软件之一。当用户使用 $\text{\LaTeX}$  系统定义的某个排版宏命令时，就能实现 $\text{\TeX}$  系统的几个或几十个基本命令才能实现的排版功能。所以现在人们往往不直接使用 $\text{\TeX}$  语言来编制排版源程序，而是用更高级的 $\text{\LaTeX}$  排版程序语言来编排西文文件。

$\text{\LaTeX}$  (读作[lah-tekh] 或[lay-tekh]) 是一种功能非常齐全的智能化西文文件排版系统。当用户在计算机上使用任何一种英文文字编辑软件以标准的ASCII 码输入期待排版的英文文件后，只须在文件的某些位置再标记些特殊的记号( $\text{\LaTeX}$  语言中的排版程序控制命令) 用以描述所希望打印出的排版样式。例如用 $\text{\LaTeX}$  语言的指令标识出哪些字句以黑体印出，从哪一行起成为新的章节，等等。输入完毕的源文件在由 $\text{\LaTeX}$  系统

软件执行排版后, 即可在计算机打印机上输出高质量的精美排版硬拷贝<sup>1</sup>。

读者也许以前使用过英文版 Wordstar, Word Processor 或其它西文文字处理程序, 但是你很快会发现 L<sup>A</sup>T<sub>E</sub>X 是一种风格迥然不同的文本软件。假如用户使用以上某种文字处理程序, 在全屏幕编辑时输入文件显示的样式与将来硬拷贝完全一样(即所谓“所见即所得”)。遗憾的是这类程序不能给出高分辨率的打印输出, 而且难以自由地调节字行空间和文件版式。目前较好的文字处理软件如 Apple Macintosh 等虽然可以获得高分辨率的打印输出, 但其智能化的程度较低。例如不能自动编排章节、段落和文献索引等, 也不能交叉引用。此外这类软件难以作版面上的重大修改和调整。L<sup>A</sup>T<sub>E</sub>X 系统则不然, 你只须在源文件中略微作些改动, 就可使文件重新排版成适合某种杂志出版的标准样式。L<sup>A</sup>T<sub>E</sub>X 能区分文件版式类型。你可告诉它所执行的文件是信件, 还是简单报告, 是文章, 还是书籍。一旦确定, 它就会自动设置页边界限, 各种标题的最佳位置和大小以及合适的字体。在排版时, L<sup>A</sup>T<sub>E</sub>X 会对所有可数的合法项目(如篇、章、节、款、项、图表以及数学公式等)逐一按标准印刷规范加上序号。当你排版的是书籍时, L<sup>A</sup>T<sub>E</sub>X 又能自动替你在目录里标上准确无误的相应的页码。L<sup>A</sup>T<sub>E</sub>X 还能自动为你标记脚注。它会对文中所引文献和文章末尾附录的文献逐一交叉配对编号。倘如你想增加或者减少所引文献, 不必担心更改文献编号引起的麻烦与差错, 因为 L<sup>A</sup>T<sub>E</sub>X 会自动为你重新编号。L<sup>A</sup>T<sub>E</sub>X 还有一些重要的特殊功能, 如制表, 作图等。

读者在使用 L<sup>A</sup>T<sub>E</sub>X 程序设计语言编辑排版源程序时, 必须记住: 如同使用其它许多电脑软件一样, 计算机毕竟是一种相当愚蠢的机器, 它可不会事先就明白你想干什么。因此你得告诉 L<sup>A</sup>T<sub>E</sub>X 你想要它为你做的每一件事情。换句话说, 你必须掌握一些最基本的 L<sup>A</sup>T<sub>E</sub>X 语言的程序设计技巧。如此方能获得令你视觉上感到满意的文件排版硬拷贝。当然, 文件排版所能达到的复杂程度与你所掌握的程序设计技巧成正比。

初次接触到 L<sup>A</sup>T<sub>E</sub>X 程序设计语言的用户常常会以为 L<sup>A</sup>T<sub>E</sub>X 是一种难以掌握的非常专业化的排版系统。然而你只要稍微解剖一下别人输入计算机内的用 L<sup>A</sup>T<sub>E</sub>X 语言编写的排版源程序, 你就会发现使用 L<sup>A</sup>T<sub>E</sub>X 并不困难。只要略微懂得一些 MS-DOS 系统<sup>2</sup> 的基本操作指令, 就能很快学会一些 L<sup>A</sup>T<sub>E</sub>X 语言中简单排版程序的设计方法。借助于 L<sup>A</sup>T<sub>E</sub>X 排版系统, 利用微型计算机终端, 你会发现非专业人员打印一篇具有专业水平, 符合国际规范, 并可直接付诸胶印或其它大规模方式印刷的西文文件并不是一件难事。

由于 L<sup>A</sup>T<sub>E</sub>X 易懂、易学, 使用方便, 它已成为当今世界上最受欢迎的西文排版系统之一。国内外许多西文杂志和书报出版社都已采用 L<sup>A</sup>T<sub>E</sub>X 程序设计语言作为标准排版语言之一。现在越来越多的学术刊物和出版社为了加快重要学术专著和论文的出版速度, 开始优先接受以软盘形式存贮的(已经用标准的 L<sup>A</sup>T<sub>E</sub>X 控制命令对论文原稿作详细排版描述, 并可由 L<sup>A</sup>T<sub>E</sub>X 系统执行排版后直接用激光照排输出)的论文小样稿。在一些发达国家里, 通过卫星通讯和计算机联网的 E-Mail 系统能在瞬息间将作者用 L<sup>A</sup>T<sub>E</sub>X 语言撰写的学术论文直接传送到杂志出版社。这样大大缩短了通常论文送审发表的周期。

<sup>1</sup>硬拷贝在本书中是指计算机排版处理后由打字机在纸上打印出的图、表和文稿清样(亦称校样)。

<sup>2</sup>MS-DOS (Microsoft Disk Operating System) 是美国 MICROSOFT 公司为 IBM PC 微机开发的系列磁盘操作系统, 也称为 IBM-DOS 或 PC-DOS。

如今, 国内许多学者也开始采用这种方式向国内外西文杂志和出版社发送稿件。

近年来, 作者与合作者一起用 $\text{\LaTeX}$ 语言直接撰写了一些英文的科技学术论文, 并且积累了一些经验。本书是作者几年来为研究生编写的选修课程的讲义, 也是自己使用 $\text{\LaTeX}$ 语言的经验总结。书中将向读者详细介绍 $\text{\LaTeX}$ 程序设计语言的结构和使用方法。通过一系列具有实用价值的排版例子的介绍, 作者希望能让读者尽可能快地使用 $\text{\LaTeX}$ 语言来为自己编排西文文章。当然本书不是一本包罗万象的关于 $\text{\LaTeX}$ 的详细使用手册。更详尽的书可参考程序设计语言的发明者——莱斯利·拉姆波特的著作 $\text{\LaTeX}$ : A Document Preparation System。

$\text{\LaTeX}$ 排版系统进行文字处理的对象主要是英文稿件, 但也能处理其它语种的文件。近几年, 排版软件市场上已出现了德英文版 $\text{\LaTeX}$ 排版系统 $\text{\LaTeX}$ 。中国的计算机工作者也已推出了中英文 $\text{\LaTeX}$ 排版软件CCT系统, 并已进入市场。书中附录部分对这两个软件中最有特色的部分作了简单介绍。

## 1.2 系统构成

### 1.2.1 $\text{\LaTeX}$ 排版系统

- 输入、编辑、排版和显示系统: IBM PC/XT、AT 微型计算机及其兼容机。屏幕显示器可为CGA、EGA、HGC和VGA等。对计算机内存的要求为640K, 硬盘容量至少为10兆, 最好是20兆以上。
- 普通宽行打印机: Brother、NEC和Toshiba等各种型号的24针打印机, Epson各种系列型号的9针或24针宽行打印机。
- 激光印字机(亦称激光打印机): 惠普(HP) II、III型或其它兼容激光印字机。

### 1.2.2 软件配置

- 可编译 $\text{\LaTeX}$ 程序设计语言的西文排版软件(例如目前进入市场的 $\text{\LaTeX}$ 各种版本的排版软件、最新推出的 $\text{\LaTeX}$ 排版系统等)。
- 支持 $\text{\LaTeX}$ 排版输出至终端的屏幕显示驱动软件。
- 支持输出至宽行打字机的打印驱动软件。
- 支持输出至激光印字机的打印驱动软件。

### 1.2.3 西文字库

- 罗马字体的白体、黑体和倾斜体、意大利斜体字、等线体、小体大写字、希腊文和规格齐全的各种特殊数学符号。
- 以上字库按分辨率区分为三种类型:

- Epson 9 针西文打印字库;
- Brother、NEC、Toshiba 和 Epson 等各种型号的24 针打印字库;
- 惠普(HP) II、III 型或其它兼容机型的激光打印字库。

所有西文字体可有十种字号供打印选择。

### 1.3 L<sup>A</sup>T<sub>E</sub>X 系统的主要排版功能

在一般排版软件中, 要想高质量地编排一篇文章、或是一本书, 可不是一件容易的事情。即使是专业编辑或是排版工作者, 有时还会被正规出版的书刊中各种各样的排版规则弄糊涂, 更不用说初学者了。但是 L<sup>A</sup>T<sub>E</sub>X 系统已事先为用户将排版过程中各种版式的规格考虑周全了, 因此用户不必再为某个排版版面的细节而白白浪费许多精力。用户只要调用 L<sup>A</sup>T<sub>E</sub>X 程序设计语言中的一些简单的控制指令, 就能实现许多非常专业化的排版功能。

L<sup>A</sup>T<sub>E</sub>X 系统的基本排版功能主要包括:

- **连字** 例如能自动将前后离散排列的字母“f”、“f”和“i”打印成联体字“ff”(无需另外添加控制命令)。
- **拼字** 系统能恰当地将字母靠拢在一起组合成看上去异常舒服的样式; 在字母与字母、单词与单词之间自动为用户选择最佳空距。
- **拆字** 在一行末尾某个单词排不下时, 系统会根据单词拆字规则将单词断开回行排版(行尾单词拆开的地方将会添上连字号“-”)。
- **行间调整** 系统会自动对不同信息量的行与行之间进行调节; 并会自动处理换行、另起段、换页等版面的调整工作。
- **行对齐** 根据控制命令, 系统可以实现居中、左齐和右齐等功能。
- **制作图表** 系统提供了功能极强的制表控制命令和分辨率极高的常用规则图形的作图工具。
- **编排数学公式** 编排各种样式的复杂数学公式(如上下标、根号、分数、矩阵、求和号、积分号等) 是系统最出色的排版功能之一。
- **章节划分** 既可自动实现章节标题和正文的规范排版, 又能记录页码信息, 从而实现目录和书眉的自动排版。
- **编排脚注和旁注** 系统能根据脚注信息量的大小自动设计脚注版面和添加脚注序号; 根据正文特殊注解的排版需要, 系统可为用户在页面左右的白边上添加旁注。
- **交叉引用** 系统可以非常方便地实现对图、表、数学方程、页面、章节、附录和参考文献等含序号的版面信息的交叉引用。



- **变换版式** 通过一些简单的参数即可实现文章、报告、书刊和信件等文体版式的变换。

## 1.4 背景知识和必要条件

本书假定读者已经熟悉微型计算机IBM-XT、AT 或兼容机MS-DOS 操作系统的基本使用方法, 并且假定用户至少拥有一个西文文本编辑软件(最好是全屏幕文本编辑软件, 例如Norton Commander 全屏幕文本编辑软件、Wordstar 编辑软件等), 并且会熟练使用该软件的各种编辑功能。唯一需要提醒的是文本编辑软件输出的文件必须是标准的ASCII 文件, 不能出现任何其它文本编辑软件体系自身的非ASCII 码特殊排版控制命令(例如行号、字体或字号处理命令等)。现在有些新版本的 $\text{\LaTeX}$  排版软件会忽略这些指令, 但是用户在未能确定你所使用的 $\text{\LaTeX}$  系统是否具有这种功能之前最好不要在源文件中留下某些文本编辑软件中特殊命令(例如Wordstar 编辑软件的排版指令)。最后建议用户常备一些质量较好的软盘, 用以拷贝编辑好的排版源文件。



## 2

# L<sup>A</sup>T<sub>E</sub>X 使用入门

本章使你学会如何制备、编译和输出一个最简单的L<sup>A</sup>T<sub>E</sub>X排版源文件。

### 2.1 基本操作步骤

要产生一个由L<sup>A</sup>T<sub>E</sub>X为你编排的可用以印刷制板的文件硬拷贝，必须要进行如下几个基本步骤：

#### 2.1.1 编辑L<sup>A</sup>T<sub>E</sub>X源文件

制备一个L<sup>A</sup>T<sub>E</sub>X排版源文件(通常称之为小样)并不复杂，只须在你所要排版的已经在计算机磁盘内建立的西文文本文件中插入一些特殊的L<sup>A</sup>T<sub>E</sub>X批处理排版命令即可。在源文件中，你可利用L<sup>A</sup>T<sub>E</sub>X程序语言来描述你所期待的文章输出时的样式。例如选取怎样的文件排版样式、字体和字号等。整个文件必须是由英文字母，阿拉伯数字，以及标准的ASCII码(American Standard Code for Information Interchange)<sup>1</sup>字符构成(即标准的ASCII码文件)。文件名可按通常的DOS文件名取法来设置，只是扩展名必须取为`.TEX`。因为L<sup>A</sup>T<sub>E</sub>X排版系统只对扩展名为`.TEX`的ASCII码文本文件进行排版处理。

#### 2.1.2 执行排版

对源文件执行编译排版是通过调用L<sup>A</sup>T<sub>E</sub>X系统的“`latex.bat`”等排版软件来实现的。系统对源文件执行排版的过程中会在计算机硬盘的L<sup>A</sup>T<sub>E</sub>X排版软件所在的当前目录里生成一个与源文件同名，但扩展名为`.DVI`的数据文件。对初学者所编的源程序来说，L<sup>A</sup>T<sub>E</sub>X或许会抱怨不懂你在文件中所输入的命令，显示屏幕上也许会出现报错信息。这时你得静下心来回到步骤1，根据出错信息重新修改你的源文件，一直到L<sup>A</sup>T<sub>E</sub>X让你通过为止。

---

<sup>1</sup>ASCII码是由美国国家标准局给常用的128个字符(字母、数字符号及控制符号等)提出的一种信息交换标准代码，简称为ASCII码。计算机输入和输出时常用这种代码。

### 2.1.3 显示校样

对过程2中所产生的DVI文件调用系统配备的显示程序,用户即可在终端屏幕上看到你所期望的排版样式(通常称之为大样,亦叫校样),其分辨率取决于计算机的显示终端屏幕。如果你对某部分不满意,则可退到步骤1再对源文件中相应的排版指令作适当修改,然后重复步骤2,直至满意为止。这样可大大减少打印输出过程中不必要的损耗,因为不管是使用激光打印机或是宽行打印机输出,其成本都是比较高的。

### 2.1.4 打印输出

经大样检查,确认准确无误后,就可调用与特定打印机适配的打印驱动软件将最后生成的DVI文件送至打印机输出。硬拷贝的分辨率同样取决于你所使用的打印机。L<sup>A</sup>T<sub>E</sub>X可支持各种分辨率级别的打印机,如通常使用的9针和24针宽行或窄行打印机。一般情况下用24针行打印机已可获得比较满意的硬拷贝。倘如用激光打印机输出,则可获得分辨率极高的硬拷贝,你既可将它放在复印机上进行大量复制,又可对其进行照相制版作大规模印刷用。

由于系统的排版过程全部是通过微型计算机来实现的,如同许多电脑排版系统一样,人们亦将L<sup>A</sup>T<sub>E</sub>X排版系统称之为高档轻印刷系统或台式排版系统。

图2.1是L<sup>A</sup>T<sub>E</sub>X排版系统的流程示意图。

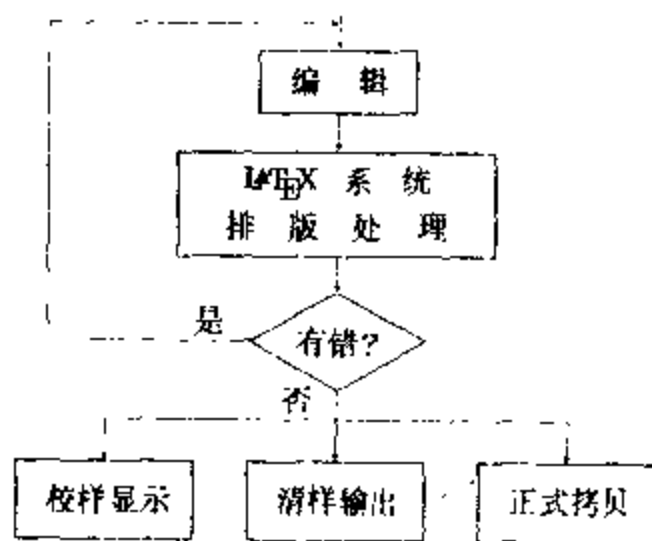


图 2.1 L<sup>A</sup>T<sub>E</sub>X 排版系统流程图

## 2.2 小样文件

图2.1中的“编辑”是指编制含有L<sup>A</sup>T<sub>E</sub>X排版命令的源程序。这是实现排版的最重要的环节之一。用户所要编辑的L<sup>A</sup>T<sub>E</sub>X排版源程序实质上是一个标准的ASCII码文本文件。在计算机排版语言中,常称之为小样文件。用户可使用手头已有的或者计算机硬

盘内配备的任何版本的中文或西文文本编辑软件来输入小样文件。例如用户可使用MS-DOS 操作系统为用户提供的行编辑程序EDLIN(一般很少使用), 也可使用中文版或英文版的全屏幕文本编辑软件如WPS、CWS 或WS。在使用WS 系列软件时建议由“N” 状态下进入编辑。有的 $\text{\LaTeX}$  西文排版系统也会为用户提供一个类似于WS 软件的全屏幕文本编辑软件。详细的使用方法可参见用户自己的系统操作手册。

为了区别不同的文件, 以便对文件修改、检索和执行排版,  $\text{\LaTeX}$  小样文件必须要有一个规范的引用名。根据MS-DOS 的规定<sup>2</sup>,  $\text{\LaTeX}$  小样文件引用名可由盘符、文件名和扩展名 .TEX 组成, 其标准格式为

[ 盘符] < 文件名> .TEX

这里盘符可以为“A:”、“B:”、“C:”和“D:”等, 输入时左右两个方括号无须输入。当使用内部命令或使用的命令在当前盘上时, 盘符可以省去。对于IBM PC、XT 和AT 系列来说, 在尖括号中的文件名可由任意不多于八个ASCII 码的字符串组成(尖括号“<” 无须输入)。例如你可将你的第一个 $\text{\LaTeX}$  小样文件取名为MYDOC.TEX 或mydoc.tex。这里引用名中英文字母的大小写对于MS-DOS 操作系统的文件检索来说完全是等价的。但是在小样文件中使用 $\text{\LaTeX}$  排版控制命令时则必须时刻注意所使用的标准指令的英文字母大小写的区别。例如在文件中插入“\LARGE”, “\Large”, 或“\large”, 则可将以后的英文稿分别以三种不同尺寸的英文大号字体编排。这里读者会发现上述三个控制令均以特殊ASCII 码字符“\”(叫做反斜杠) 开头。反斜杠字符在这儿起到了引导排版命令的作用, 能被 $\text{\LaTeX}$  系统所识别, 并能按照程序设计者的要求, 根据后面叙述的指令对小样文件执行排版。像这种能被识别并能使系统执行某种特定功能的字符称之为控制字符。除了几个特殊的控制字符(例如: \$, % 和 & 等) 之外, 几乎所有的 $\text{\LaTeX}$  控制命令均以反斜杠开头。例如在排数学公式时若要打印出希腊字母“ $\gamma$ ”和“ $\Gamma$ ”, 则可分别通过控制令“\gamma”和“\Gamma”来实现。又如打印“ $\text{\TeX}$ ”和“ $\text{\LaTeX}$ ”专用记号是由命令“\TeX”和“\LaTeX”分别控制打印的。

小样文件中的正文内容可多可少, 具体的控制令方式根据文稿的排版要求可简可繁。然而任何一个可被执行排版的小样文件至少要有如下三个排版控制命令:

### 2.2.1 文件版式命令

所谓“版式”, 是指被编辑的论文、报告或书籍等各种类型的文件正文部分的全部格式: 包括各级标题所占空间及修饰, 正文、表格、图文、脚注等的字号和字体以及行空规格, 版心的大小和边空尺寸, 排版要求以及上述各方面的相互配合等。

为了对排版的文稿按版式分类,  $\text{\LaTeX}$  源程序(被编辑的小样文件) 的第一行必须输入一个带若干参量的文件版式控制命令, 其标准格式为

\documentstyle [版式注解1, 版式注解2, ...] { 版式类型 }

<sup>2</sup>张福炎等编著, 《微型计算机IBM PC 的原理与应用》, 南京大学出版社, 1990 年版

这里参量“版式注解1, 版式注解2, ...”可分别对文件版心正文英文字体及字号的选择、版心是排成单栏还是双栏的选取, 页边距的宽窄形式设定等作详细说明。省略这些参量中的某一个或全部时, L<sup>A</sup>T<sub>E</sub>X 将为你按照系统内部设置的标准版式进行排版。下面是几个常用的可供选择的版式注解:

**11pt** 将版心正文英文字体(基准字)的大小相对标准字体放大10%。如果用印刷术语“点”(point)<sup>3</sup>来衡量版心正文英文字大小, 则标准字体为10点, 其尺度相当于一般中文刊物中版心正文所采用的标准五号铅字的大小。而这里参数“11pt”即置基准字体为11点。

**12pt** 正文英文字体的大小相对标准字体放大20%。

**twocolumn** 正文版面将按照通常学术刊物分成两栏排版; 省略时按通栏(单栏)样式排版。

**twoside** 正文版面将按照书籍样式双面排版; 无此参数时系统一般按单面版式排版, 对于版式类型参量定义为书籍类(book)特殊情形时, 系统将自动设置双面排版模式。

**titlepage** 对于标题、摘要所处的版面另页排版。此参量仅适用于版式类型参量为文章类(article)的文件。

另外还有一些可供选择的版式注解, 在以后的一些章节中将陆续作一些介绍。各种新版本的L<sup>A</sup>T<sub>E</sub>X排版系统还会提供许多特殊的版式注解, 用户可通过查阅与系统配套的操作手册来使用这些版式注解。

版式类型参量主要用以描述执行排版的文体形式。可供选择的参量主要有文章类(article)、报告类(report)、书籍类(book)和信件类(letter)等。文章类(article)和报告类(report)两种版式之间的区别仅在于内部章节的划分方式。前面的版式注解参量可以省略, 但是这里的版式类型参量必须标明, 并且只能选取一种版式类型参量。所以你必须要在源文件中注明你要排版的文体是属于文章类还是报告类, 是属于书籍类还是其它类型。可供选择的版式类型参量的多少亦取决于你所使用的L<sup>A</sup>T<sub>E</sub>X软件包。一旦确定文体形式, L<sup>A</sup>T<sub>E</sub>X就会根据相应的文件版式来进行排版。附录D中给出了一些最基本的常用版式类型参量。

### 2.2.2 文稿排版环境起始命令

设定文件版式控制命令后, 用户还可以再对排版文件的全局作一些注解。例如行距, 版面的大小尺寸, 页边距等。这些注解涉及到一些比较复杂的排版技巧。幸运的是用户可以暂且不用去管这些命令使用方法的具体细节, 因为系统已为用户设置了多种在省略方式下最简单的标准排版形式, 所以用户只须知道如何输入文稿的方式就可以了。

在输入正式文稿之前首先得为源文件设置一个必不可少的文稿排版环境, 所以你还得再加上一行文稿起始命令。其形式颇有点像其它一些计算机高级程序语言: 例如Pascal、Turbo Pascal等语言。文稿起始命令的格式为

<sup>3</sup>1点(pt) =  $\frac{1}{72.27}$ 英寸 = 0.3515毫米

```
\begin{document}
```

它的作用是启动系统开始对此命令后的正文文稿进行编译排版。

### 2.2.3 文稿排版环境结束命令

在源文件中输入第二行文稿起始命令之后,你就可以从第三行起按照通常打字习惯由键盘随意输入正文以及相应的排版控制命令。但是在正文输入工作完毕之后,你还必须要再添上一行文稿排版环境结束命令:

```
\end{document}
```

其作用是告诉系统立即无条件结束文稿排版工作。在此命令以后的任何文稿和控制命令将被忽略。在具体编辑文件时还可通过在小样文件中移动此命令的位置来中断排版工作的进行,从而分解排版中遇到的困难,以便及时发现和排除源程序中的语法错误。

如果不是涉及比较专业化的复杂文件,那么有了上述三个排版控制命令,你就可以编辑比较简单的 $\text{\LaTeX}$ 小样文件了。下面是一个可被 $\text{\LaTeX}$ 系统执行排版的简单源程序实例:

```
\documentstyle{article}
\begin{document}
This is a first example of simple input file.
\end{document}
```

$\text{\LaTeX}$ 系统执行排版后,会输出

```
This is a first example of simple input file.
```

这里读者可以看到配对的开启和关闭控制命令“\begin”(起始)和“\end”(结束)实际上设置了一个完整的正文文稿(document)的排版环境。类似的由“\begin”和“\end”构成的设置其它排版环境的 $\text{\LaTeX}$ 开关控制命令还有许多。在后面的各章节中将会对常用的排版环境命令作详细介绍。

## 2.3 打字与排版入门

简单的英文正文部分不外乎是通过像打字机一样的键盘输入的一连串的单词、特殊字符、句子、标点符号、段落和用来隔开这些字符串便于区分的空格构成。你在输入这些内容时不必像在打字机上那样把过多的精力耗在考虑字与字以及句子与句子之间空格的多少、行与行之间距离的大小、以及换行和英文单词拆字的位置上面, $\text{\LaTeX}$ 会按西文书刊印刷的国际规范为你排版。

$\text{\LaTeX}$ 有一些根据通常西文书写和打字习惯的隐含排版规则,这些规则会令用户感到使用非常方便。下面我们先介绍其中最基本的几条规则。



### 2.3.1 空格

L<sup>A</sup>T<sub>E</sub>X 在读到你由键盘(标准键盘上的长形空格键)输入的隔离字与字和句子与句子的空格符号(本书用“`□`”表示)后<sup>4</sup>, 会为你根据版式定义的一行所能容纳的信息量的大小来自动设置字、词和句后所应留出的最佳空间。在输入空格时, 字词间额外多打了几个空格键无所谓。换句话说, 在小样文件中两个英文单词之间留一个空格与十个空格对于 L<sup>A</sup>T<sub>E</sub>X 来说其功能完全等价。行首或行尾的额外空格(例如用 Tab 键输入的信息), 以及由回车键(标准键盘上的 ENTER 或 RETURN 键)打入的强迫换行记号对实际排版来说, 只等价于一个空格符号。L<sup>A</sup>T<sub>E</sub>X 在排版时会自动换行, 并会将该行内容根据版心的左右边距要求两边撑足。你在源程序中输入的正文文稿中的各行也许会长短不一, 参差不齐, 但是这不要紧, 因为对于 L<sup>A</sup>T<sub>E</sub>X 来说这些在实际排版中毫无影响。为了便于在屏幕上校对和改错, 建议在输入和编辑小样文件时养成每行输入的 ASCII 码字符数不超过 72 个的习惯(有些接收以 L<sup>A</sup>T<sub>E</sub>X 源文件形式的文章或书稿的出版社会对作者作行字符数上限的规定)。在一般的全屏幕文本编辑软件中, 小样文件中的换行可由回车键来实现。

### 2.3.2 换行

有时你会希望文稿排版到某处时立即另起一行从头再排。这时, 文字内容可能还未排到版心的右边界处, 即版面的右面可能无法与其它行拉齐。如果用户想要 L<sup>A</sup>T<sub>E</sub>X 为你换行, 那么只要在需要换行的位置连续插入两个反斜杠控制符“`\\`”即可。L<sup>A</sup>T<sub>E</sub>X 在读到控制符号“`\\`”时, 便会立即无条件结束在当前行上的排版工作, 并将文稿后面的内容从下一行顶格起排。

与双反斜杠换行控制符“`\\`”等价的控制命令是“`\newline`”。你可以在需要换行的位置插入此命令即可。

### 2.3.3 分段

如果用户在一行文字输入完毕后想要编排另起一段的文字的话, 则只要在小样文件中再额外空出一行即可。这可以由连续两个回车键来实现。当 L<sup>A</sup>T<sub>E</sub>X 读到一个以上的空行记号时会自动将后面的文稿另起一段排版, 并且会在新的一段的首处缩进适当位置排字; 此外还将在两段之间留出适当空白间距。与空格记号相似, 在小样文件中两个段落之间留一个空行与留十个空行的功能完全等价。如果你不喜欢采用在源文件中额外再空出一行来实现分段的方法, 那么你还可使用另外一种等价的排版控制命令“`\par`”。这种命令使用很容易, 用户只要将其插入需要分开的两段文稿之间就可了。这时你就用不着再在文稿中另外空出一行了。

---

<sup>4</sup>空格符号是指标准的 ASCII 码空格符(在 CCDOS 中西文操作系统中, 不能用全角的中文空格符, 否则会出错, 或达不到预期的排版效果)

### 2.3.4 换页

换页是指文稿中某个篇、章等内容在某页上排完后不接排下一个篇、章，而是另起一页排版。在印刷术语中，换页分另起页和另起面两种。另起页是指新页码总是保证为单数(若当前页为单数，则用空白页补足)，适用于双面排版情形(如本书排版样式)；另起面则不分单双，新页码总与当前页连续。在 $\text{\LaTeX}$ 源程序中，需要另起面时，可插入排版命令“ $\backslash\text{newpage}$ ”或“ $\backslash\text{clearpage}$ ”；而另起页则可使用“ $\backslash\text{cleardoublepage}$ ”。

### 2.3.5 拆字

$\text{\LaTeX}$ 在排字接近每一行行尾处时总会遇到各种各样、长短不一的英文单词。在机械式打字机上，拆字是令打字员头痛的一件事。但是在 $\text{\LaTeX}$ 排版系统里你就不必为此劳神了。因为 $\text{\LaTeX}$ 会为你安排妥贴。假如遇到较短的单词， $\text{\LaTeX}$ 会在保证行宽不变的前提下自动调整字词间距设法将其排入当前行。倘如失败，则会将其放到后面一行顶格处，同时将当前行按版心行宽撑足排版。假如遇到较长的单词， $\text{\LaTeX}$ 还会根据标准的英文单词的行尾拆字规则自动按音节将单词作拆字处理后再行排版，并会记得为你在单词的前半部分添上连字号“-”。偶尔会遇到某些专业单词无法按一般规则进行正确的拆字处理，这时 $\text{\LaTeX}$ 会在显示屏幕上向你发出求援信息。遇到这种情况用户可以采取置之不理的方针，但也可通过人工干预来处理拆字。你可在你认为是允许将单词断开的正确位置插入若干个控制符“ $\backslash-$ ”。 $\text{\LaTeX}$ 在执行排版时会选择用户插入的几个指令中最合适的一个来处理拆字，而该行中的其它多余的控制符“ $\backslash-$ ”则会被忽略掉。

### 2.3.6 连字

所谓连字是指在正规印刷物中出现的诸如“ff”和“B”之类的由几个单独字母联体构成的字符。在某些排版系统中，实现连字通常是采用比较复杂的控制命令来实现的。在大规模的排版源文件中，成千上万次的为了实现简单连字的控制命令的输入是令人非常厌烦的。对于常规的连字，在 $\text{\LaTeX}$ 排版系统中，你就根本不必为此犯愁。因为系统在执行排版时会自动将该联体的离散字合成一个连字。你实际上对此一窍不通都没有什么关系。

### 2.3.7 标点符号

关于标点符号有如下几点需要说明。

① 合法标点符号  $\text{\LaTeX}$ 默认在小样文件出现的如下16个ASCII码字符为合法标点符号：

. , : ; ? ! ' ' ( ) [ ] - / \* @

系统认定源程序中字符“.”、“?”和“:”后面出现一个以上的空格时，即视作完整的句尾标点符号，排版时会额外引进一些空距。不过如果西文句子结尾出现大写字母时则不然，用户得在标点符号之前插入控制符“ $\backslash\text{e}$ ”作为结尾标记。例如源程序

This is Sentence I\@. Note the sentence-ending period  
follows an upper-case letter.

输出结果为:

This is Sentence I. Note the sentence-ending period follows an upper-case letter.

② 引号 注意由计算机标准键盘上的双引号键输入的字符(") 在上面列举的标点符号行中没有地位。L<sup>A</sup>T<sub>E</sub>X 输出校样中的单、双引号的排版全部是由单引号(单上引号(') 或单下引号(')) 来控制完成的。L<sup>A</sup>T<sub>E</sub>X 在编译小样文件时会将单个出现的单上引号或单下引号字符视作标点符号中的单上引号或单下引号来进行排版。如果你需要在输出时打印出双引号, 则可分别通过连续输入两个相同的单引号字符来控制产生。例如源文件中连续出现的两个单上引号('') 可输出一个双上引号(""), 而连续两个单下引号('') 则可输出一个双下引号(")。

③ 破折号 L<sup>A</sup>T<sub>E</sub>X 语言里没有专门的类似中文破折号的专用字符, 但是你可通过连续键入若干个减号(连字符) 来标识不同长度的破折号。L<sup>A</sup>T<sub>E</sub>X 会自动为你将几个减号连起来成为一个破折号。根据西文印刷物的国际规范, 通常有三种不同功能的破折号:

- 连字符 对于换行拆字时或几个西文单词连成的复合单词中所用的连字符, 可用计算机键盘上的减号键“-”来控制打印。例如源文件程序“man-made”能打印出“man-made”。
- 起迄符 起迄符通常用于表示数字范围(“从……到……”), 它可用两个减号“--”来标识。例如小样

00010--00999

可以输出拷贝“00010-00999”。

- 西文破折号 破折号通常用于句中注解, 可通过连续三个减号“---”字符来生成。例如源文件

\LaTeX{} --- A Document Preparation System

输出为

L<sup>A</sup>T<sub>E</sub>X — A Document Preparation System

④ 省略号 省略号由专门的打印命令来控制。命令“\ldots”将为你输出接近本行底线的三点形省略号“...”, 而在数学公式状态排版环境里命令(关于数学公式排版环境的内容可参见第六章)“\cdots”则会为你在公式中输出上下居中的省略号“...”。

### 2.3.8 控制字符

在 $\text{\LaTeX}$ 系统里除了反斜杠符号“\”能起引导排版控制指令的作用之外, 还有如下九个有类似特殊排版功能的 $\text{\LaTeX}$ 控制字符:

# \$ % & \_ { } - -

$\text{\LaTeX}$ 将这些字符与反斜杠符号“\”一起定义为系统的保留字符, 其用途将在后面的章节中陆续作详细介绍。要想减少文件编辑中的错误, 读者须非常谨慎地使用这些控制字符。由于在西文文件中有时免不了需要打印前面七个符号, 为此,  $\text{\LaTeX}$ 设计了一种能在输出校样中打印出这些具有特殊功能的控制字符的简易打印命令。假如你想在输出文件中打印前面七个字符中的某一个, 则可通过在所要打的字符前简单加上一个反斜杠控制符来实现:

\# \\$ \% \& \\_ \{ \}

至于最后两个则稍麻烦些, 打印控制令分别为:

\^{} \\_{}

面要想打印反斜杠符号“\”的话, 则须输入命令:

$\$ \backslash \$$

到此为止, 读者差不多已获得了产生一个简单 $\text{\LaTeX}$ 西文小样文件的初步知识。下面的问题是如何让 $\text{\LaTeX}$ 系统来编译你的源文件。

### 2.3.9 源程序注释控制符

对使用者来说, 有时需要在 $\text{\LaTeX}$ 排版源程序中的适当位置加上一些注释语句, 希望既能提高程序的可读性, 又不影响源程序的排版功能。如同许多常规的程序设计语言一样,  $\text{\LaTeX}$ 采用ASCII码中的百分号“%”作为注释控制符。用户在源程序中的任意一行中插入注释控制符后, 系统在执行排版时即会忽略该行后面一部分的排版信息, 而转向源程序中的后续行执行排版(每个注释控制符仅在源程序中的一行内生效)。

如果用户想打印出百分号“%”, 那么必须在源程序中字符“%”后面缀上反斜杠控制符“\”。否则的话, 不仅不能打印出百分号, 还会在排版时丢失源程序中位于字符“%”后面至行尾的文字内容。

## 2.4 执行排版

假定你工作的当前目录是处在硬盘C: 中的 $\text{\LaTeX}$ 的子目录下, 并且已经输入了如下所示的取名为CHO2EX02.TEX的小样文件,

```
\documentstyle[12pt]{article}
\begin{document}
```

This is a second example of simple \LaTeX{} input file.

The end of words and sentences are marked  
by spaces. It doesn't matter how many  
spaces you type; one is as good as 100!

One or more blank lines denotes the end  
of a paragraph.

'Conversaition' dictates that punctuation go  
inside quotes, like ''this,' ' but I think  
it's better to do ''this''.

The seven special characters

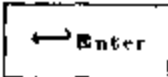
( \# \\$ \% \& \\_ \{ \} )

are easily produced by typing a backslash symbol in  
front of the corresponding character.

\end{document}

那么就可以让L<sup>A</sup>T<sub>E</sub>X 为你执行排版了(注意在小样文件第三行中我们引入了打印特殊符号的控制令\LaTeX, 其输出效果就是读者到现在为止看到许多次的“L<sup>A</sup>T<sub>E</sub>X”专用记号)。在西文MS-DOS 操作系统中, 当处在PCTEX 目录下(此时计算机屏幕上显示待命符号c>或C:\PCTEX>) 时, 你就可以在键盘上打入如下一行命令:

C:\PCTEX>latex ch02ex02 ✓

行尾的记号✓ 表示键盘上的回车键  , 一旦按下此键, L<sup>A</sup>T<sub>E</sub>X 就会对文件 ch02ex02.tex 执行排版<sup>5</sup>。注意在上面命令中已经省略了扩展名[.TEX], L<sup>A</sup>T<sub>E</sub>X 会自动为你搜寻扩展名为TEX 的ch02ex02 文件。当系统接受你的指令后, 屏幕上会依次显示出一些运行过程中的有关信息。例如

```
C:\PCTEX>Personal TeX, Inc 1985, 1986. Serial # 1111
This is TeX, Version 2.0 (PCTeX 1.50, (c)Personal TeX, Inc 1986)
(preloaded format=lplain 92.3.4)
**&lplain ch02ex02.tex
(ch02ex02.tex
LaTeX Version 2.09 - Released 18 December 1985
(\pctex\texinput\article.sty
Document Style 'article'. Released 17 December 1985
(\pctex\texinput\art11.sty))
No file ch02ex02.aux.
```

<sup>5</sup>如果是VAX 机, 则命令应改为\$**LATEX** CH02EX02 ✓。

```
{1} (ch02ex02.aux))
```

```
Output written on ch02ex02.dvi (1 page, 956 bytes).
```

```
Transcript written on ch02ex02.log.
```

第一行到第四行是告诉用户 $\text{\LaTeX}$ 所调用的 $\text{\TeX}$ 排版系统软件的版本。 $\text{\TeX}$ 是 $\text{\LaTeX}$ 系统中最重要基础排版软件。它本身可构成一个完整的非常专业化的独立排版系统。有关这方面的知识已超出了本书的范围,读者可暂且不去管它。当然在 $\text{\TeX}$ 为你编译报错时,懂点 $\text{\TeX}$ 系统的出错信息ABC是会对你大有裨益的。我们将会在本书有关出错信息的章节中作一些简要介绍。

第五行和第六行是告诉用户小样文件ch02ex02.tex已搜寻到并且正在被 $\text{\LaTeX}$ 系统读进。第七行描述了你所使用的 $\text{\LaTeX}$ 的版本。接下去的三行则罗列了在编排文件类(article)类型文件时所调入的其它各种各样的系统应用软件。

至于“No file ch02ex02.aux”是一行警告信息。只是告诉你在这之前你并未令 $\text{\LaTeX}$ 对源文件ch02ex02.tex执行过排版。 $\text{\LaTeX}$ 在对每个小样文件执行排版时,会产生一个引用名与源文件名相同但扩展名为[AUX]的重要信息拷贝文件。在这文件中将记录有关章节目录、参考文献和图表等信息。当再次执行排版时这些信息就会被查询。而第一次排版时的警告信息消失而转为显示所搜寻到的相应的AUX文件。

在上面倒数第三行头上以方括号形式显示的号码表示已经为用户编排好的可输出校样的页码。而后面的内容则是报告新的版本的AUX信息拷贝文件已经产生。如果你在正文中改动了某些章节和引用的文献,相应的新的自动调整的页码和编号又会被记忆在AUX文件中,因此你完全不必为此担心。

最后的两行是与输出有关的信息。前面一行是告诉用户排版结果已以扩展名为[DVI]而引用名同小样文件相同的文件形式存贮在当前工作目录下。DVI文件(Device Independent File的简称)是一种与具体打字机无关的排版信息数据文件。使用支持特定打字机的设备驱动软件,你就能获得可供印刷的文件校样。在文件名ch02ex02.dvi后面括号里的内容是可生成的文件校样的页数以及DVI文件在MS-DOS系统内存所占的字节(bytes)数。末尾的一行是让你知道在执行排版过程中屏幕上显示的和系统内部的有关信息全被拷贝到可读的ASCII码文件ch02ex02.log里了。

如果你的计算机旁配备了行式打印机或其它类型的输出硬拷贝的设备,你就能在A4复印纸<sup>6</sup>上看到如下打印结果(用户也可在计算机终端显示屏幕上看到模拟输出的结果):

```
This is a second example of simple  $\text{\LaTeX}$  input file.
```

```
The end of words and sentences are marked by spaces. It doesn't matter how  
many spaces you type; one is as good as 100!
```

```
One or more blank lines denotes the end of a paragraph. 'Conversition' dictates  
that punctuation go inside quotes, like "this," but I think it's better to do "this".
```

```
The seven special characters { # $ % & - { } } are easily produced by typing a  
backslash symbol in front of the corresponding character.
```

<sup>6</sup>A4复印纸是一种最常用的复印机用纸,标准尺寸为长297毫米,宽210毫米。若以英寸为长度单位则长为11.75in,宽为8.25in。

注意与实际打印结果略有差别的是, 这里为了节省版面空间将A4纸空白部分和底部中间的页码“1”去掉了。

## 2.5 排版输出

当小样文件经过L<sup>A</sup>T<sub>E</sub>X排版处理之后, 系统有如下三种基本的校样输出方式供用户选择: 屏幕显示、宽行打印机输出和激光印字。第一种方式是排版过程中在屏幕上的模拟显示, 多用于未定稿前的校样预检, 是一种最廉价的无损耗输出。第二种方式一般用于排版定稿后输出。这种方式只是消耗了一些纸张和色带, 是一种比较经济的打印输出方式。倘若打印机是24针型的, 那么在一般场合下使用深色色带打印的排版拷贝已足够好了。第三种方式输出的硬拷贝最漂亮精致, 但是要使用价格比较昂贵的消耗碳粉的激光打印机, 所以一般仅在最后定稿并用于正规印刷的情况下时才采用此方式输出拷贝。下面我们对每一种输出方法作简要介绍。

### 2.5.1 排版样式的屏幕显示

L<sup>A</sup>T<sub>E</sub>X系统提供了一个适用于不同分辨率屏幕(如CGA, EGA)的可执行排版样式显示软件view.exe。当处在PCTEX子目录下时, 有两种方法可进入显示系统。一种是由键盘输入命令view即可进入系统, 然后再通过指令输入文件名; 另一种是由键盘输入带文件名参量的命令:

```
view [DVI 文件名] /
```

进入系统。例如我们可以键入

```
view [.exe] ch02ex02[.dvi] /
```

(注意指令中方括号内的部分可以省略), 这时屏幕上会出现一个如图2.1所示多项目选择菜单。最上面的两行英文字给出了显示屏幕分辨率以及软件版本的有关信息。中间的窗口是一个具有11种功能的选择菜单。最下面两行则是一个菜单使用的简要说明。

屏幕上可选的每一项都是通过键入该项英文名字的第一个字母(屏幕上以大写形式显示)来实现参量修改的功能的。当打入你要选择修改项对应的字母(大小写均可)后, 该选择项便被激活。你就可非常容易地实现参量的修改。该选择项修改结束后按回车键即能生效。下面对显示系统菜单上的每个选择项作一介绍。

- **File(文件名)** 所要显示的DVI文件名。按下计算机键盘上的 **F** 键则立刻会激活文件名选择项, 这时在File: 后面便出现一行闪烁着的等待你输入信息的下划线字符(共40个)。此时你即可输入你需要观看的文件名(扩展名[.DVI]可省略), 然后再按回车键后你的文件便被系统调入。假如文件名打错或系统不存在要显示的[.DVI]文件, 则在选择菜单方框底部的中间区域会出现一行“DVI file not found”(DVI文件未找到)警告信息。这时你或者可按 **ESC** 键退出显示系统, 或者可再按











MAXview. IBM Enhanced Graphics, Color EGA (640x350)			
V:2.j 1986 (c) Aurion Tecnologia SA de CV			
File: ch02ex02.dvi			
Begin on page:	-30000	Height:	11.0
End on page:	30000	Width:	8.5
		Top offset:	1.0
Magnification:		Left offset:	1.0
Screen reduction:	2	Y (top) marg:	1.0
		X (left) marg:	.5
Press first letter to select an item.			
<Return> to start processing, <Esc> to abandon.			

图 2.2 显示系统的选择菜单

**[F]** 键重新输入另外一个文件名。假如在进入显示系统之前已经在view命令后面加上了正确无误的文件名参量, 则此选择项可不必再调用。

- **Begin on page**(开始页页码) 当文件校样稿很长, 页数较多时可控制屏幕最先显示的模拟排版后的页码。此项省略时系统为你首先显示排版后的第一个版面。
- **End on page**(结束页页码) 希望模拟显示的最后一个版面对应的页码。在页数较少时本项和第②项不必调用。省略时能显示的下限是你排版后的最后一个版面。
- **Magnification**(放大) 放大显示。除非你要研究放大图像的具体细节, 一般不要调用此项。在微机 $\text{\LaTeX}$ 系统(PCTEX)里, 特大号字体未安装进去, 因而般大项目无效果。
- **Screen reduction**(缩小显示) 有三种尺寸的模拟版面图像可供选择。重复按**[S]**键, 则缩小的倍数参量会自动在1,2和3之间变换。3对应的图像与实际比例为1:1, 但不清晰; 1对应的图像字体比较大。此项省略时系统设置在较适中的参量2状态下。
- **Height**(纸高) 键入**[H]**可改变在屏幕上模拟输出的虚拟纸张的长度。选择省略时屏幕上设置的是标准A4复印纸的长度(本项目和以下各选择项中的度量均以英寸为单位)。对于一般显示要求来说, 此项无须更改。
- **Width**(纸宽) 键入**[W]**可进行模拟输出的虚拟纸张宽度的更改。原来设置的是A4复印纸的宽度。此项亦不必更改。

- Top offset(顶端边距) 此项可用于设置版心与虚拟纸张顶部的边距。省略时系统设置的纸边距为1英寸。
- Left offset(左边距) 用于设置版心与虚拟纸张左边界的边距。省略时系统设置的左边距为1英寸。
- Y (top) marg.(距版心上边界位置) 用以调节虚拟翻阅每一页内容时一开始显示在屏幕左上方的图像在实际版面上距版心Y正方向上边界位置, 省略时系统设置的参数为1英寸。
- X (left) marg.(距版心左边界位置) 调节显示在屏幕左上方的图像在实际版面上距版心X负方向上边界位置, 省略时系统设置的参数为0.5英寸。


用户在选择了要显示的排版文件名、显示版式的缩小方式、开始页码和结束页码后再按回车键即能在屏幕上看到放大一定倍数的模拟排版结果。由于屏幕窗口视见区有限, 不可能整版显示。但是用户可通过在键盘上的四个位移控制键 、、 和  来移动整个版面。要向后(朝页码大的方向)翻页时可按 -Enter 或  键, 而倒过来翻页时可按  键。通常情况下系统为你是按反视(白字黑底)状态显示版面的, 你可通过撤减号键实现由反视到正视(白底黑字)或相反的转换。任何时候要退出显示系统可按  键。

## 2.5.2 宽行针式打印机输出

L<sup>A</sup>T<sub>E</sub>X 可支持多种类型的针式宽行打印机输出。国内比较流行的打印机: 如NEC、Brother、Epson LQ 系列和Toshiba 等各类24针系列打印机均可在L<sup>A</sup>T<sub>E</sub>X 系统下使用, 甚至对于Epson 9针系列, L<sup>A</sup>T<sub>E</sub>X 亦能给出较好的拷贝。不过在使用时, 还须在系统配备支持具体打印机的驱动软件。

打印输出的具体操作指令将取决于你拥有的软件 and 打印机的类型。例如对于Epson LQ1500 及其它LQ 系列的各式宽行打印机, 则可在排版系统中很容易地找到有关的可执行打印驱动软件。目前在国内使用比较广泛的PCT<sub>E</sub>X 软件中配置的打印驱动软件为“DVIELQ.EXE”, 使用时可以键入命令:

```
dvielq (DVI文件名) /
```

一旦按下最后一个回车键 -Enter, 打印驱动软件就会被调用。如果输入命令时没有指明被打印的DVI 文件名, 则屏幕上会出现请求输入DVI 文件的等待信息:

```
C:\PCTEX>dvielq
```

```
DVIFNAME: _
```

```
(DVI 文件名: )
```

当用户输入文件名后, 驱动程序会在屏幕上陆续打印出一些关于打印机接口和打印页码的选择项。对于仅有几页纸的小规模文件, 用户只须连续按下回车键即可将DVI文件输出至打印机上。在入机对话时用户可选择的项目有如下三个:

1. Printer file name [prn]: (打印设备文件名)  
通常只要按回车(省略状态)即可。省略时系统调用MS-DOS系统的PRN(等价于LPT1)设备文件。
2. Starting page (default=\*): (起始页码)  
如果用户不键入所须打印的起始页码, 而是按下回车键(省略状态), 则系统自动选择从文件第一页开始输出打印。
3. Maximun number of pages (default=1000000): (打印总页数)  
如果用户不键入所须打印的页面总数, 而是按下回车键(省略状态), 则系统自动选择打印至文件的最后一页时才停止。

假定用户的打印机接在串行输出设备(COM1)上, DVI文件名为mydoc.dvi, 且只须打印第20页至第88页, 则打印命令的输入方式如下:

```
C:\PCTEX>dvielq mydoc ✓
DVIELQ version 1.1 (c) Personal TeX Inc. 1985
Printer file name [prn]: com1 ✓
Starting page (default=*): 20 ✓
Maximun number of pages (default=1000000): 69 ✓
.....
```

### 2.5.3 激光打印机输出

L<sup>A</sup>T<sub>E</sub>X系统配备的高分辨率打印驱动软件可支持日前进入市场的各种型号激光打印机(如近年来国内用户使用比较多的激光打印机: HP Laserjet (惠普) II、III型, Canon等系列)。一般系统配备的是HP系列打印驱动软件(与其它一些激光打印机兼容)。例如在PCT<sub>E</sub>X排版软件中配有

TEXP.BAT	(激光打印批处理文件)
PTIHP.EXE	(生成激光打印数据文件的驱动程序)
PTISPOOL.EXE	(激光打印驱动软件)

L<sup>A</sup>T<sub>E</sub>X执行排版以后生成的DVI文件虽然记录了有关排版细节的全部信息, 但还不能直接在激光打印机上输出。打印时首先要将DVI文件翻译成激光打印的数据文件, 然后才由打印驱动软件输出至激光打印机。如果用户不准备对打印输出的具体参数(如页码数、纸张大小、拷贝份数等)作修改, 则可使用系统配备的打印批处理文件。这样可将以上两个过程的打印工作通过下述操作指令一次完成,

texp <DVI文件名> ✓

假如用户希望对打印参数作些更改, 则可先调用生成激光打印数据文件的驱动程序。对于PCT<sub>E</sub>X排版软件, 在操作指令

ptihp <DVI文件名> ✓

发出之后, 整个屏幕上会立即显示一个如图2.3所示的选择菜单。

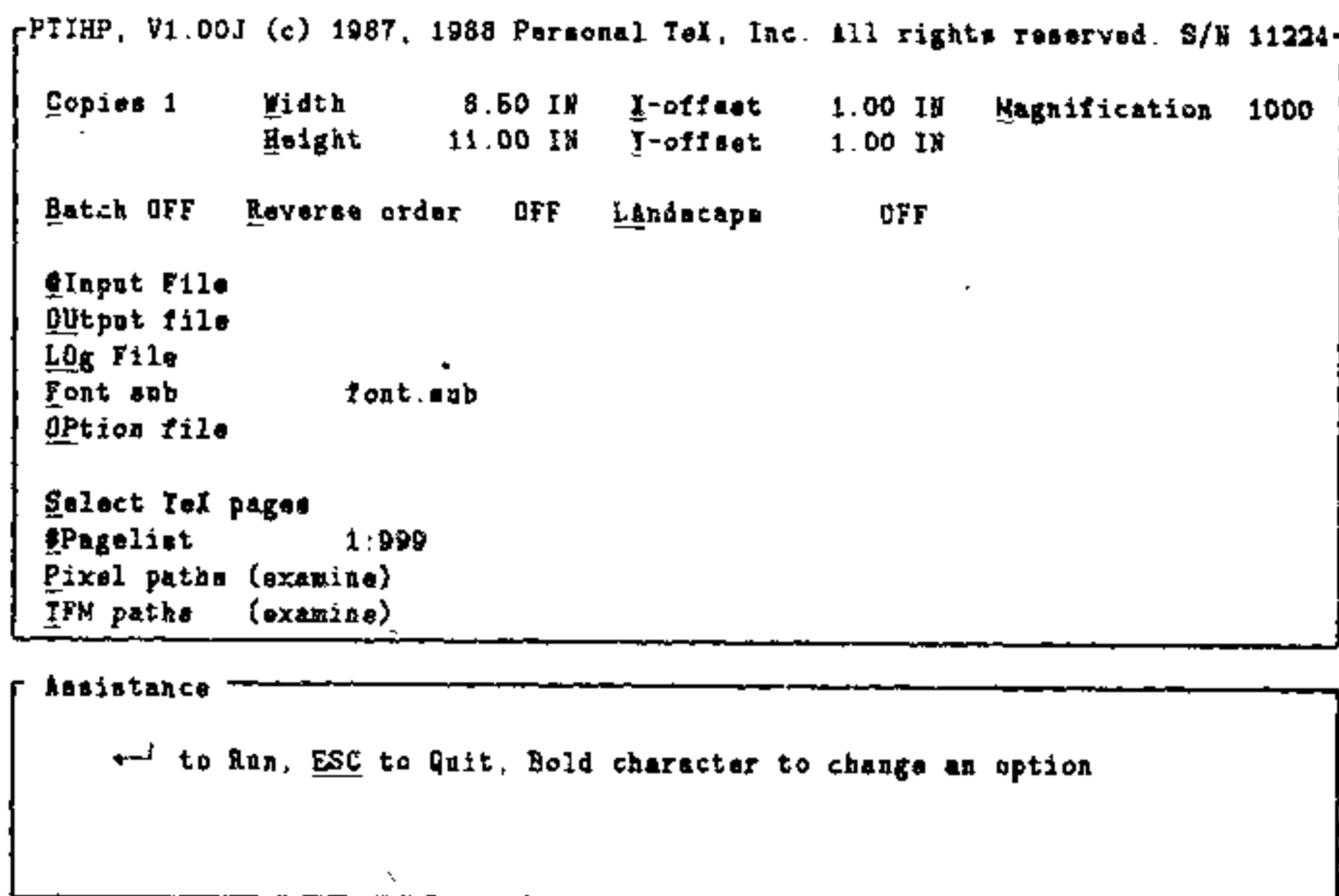


图 2.3 激光打印驱动程序(PTIHP.EXE) 屏幕显示菜单

激光打印驱动程序(PTIHP.EXE) 的屏幕显示菜单分成上下两个部分。其中上面一部分是可激活的选择菜单, 而下部的则是系统提供帮助的信息显示窗口。菜单中可选的每一项都是通过键入该项目标题的前面一个(或前面两个)字母(或字符)(见图2.3中用下划线标记的字母或字符)来实现参量的选择。当键入你需要修改的选择项所对应的字母(大小写均可)后, 该选择项便被激活。这时, 屏幕中央立即会弹出一个可输入数据或确认运行状态的选择子菜单; 同时在下部的帮助窗口中又为用户显示一些有关选择参量输入方式的信息。于是你就能像平常修改文本文件那样, 通过使用键盘上的一些功能键来实现参量的修改。每个选择项修改结束或状态确定后按回车键即能生效。下面对惠普激光打印驱动系统显示菜单上的每个选择项作一简单介绍。

- Copies(拷贝份数) 所要重复打印输出的文件份数。按下计算机键盘上的 **C** 输则立刻会激活拷贝份数选择项, 这时在屏幕中央会弹出一个选择拷贝份数的小型子菜单。在菜单方框内有一个闪烁着的等待你输入数字的黑色光斑。此时你即可输入你需要文件拷贝份数(注意不要超过99), 然后再按回车键。此项不选择时(省略状态)系统自动设置拷贝份数为1。如果激活拷贝份数选择项后并不想作任何修

改, 则只要再按一下回车键即可退回到原来打印系统的静态屏幕显示菜单。

- **Width(纸宽)** 键入 **W** 即可在弹出式样窗口中修改打印输出时所使用的纸张宽度(长度单位可以是 $\text{\LaTeX}$ 语言中合法的绝对单位: 如in、cm、pt等, 以下各选择项中的度量单位规则相同)。宽度大小限制在1到19 in之间, 省略时系统将其设置为A4复印纸的宽度。此项一般不必更改, 因为通常打印机使用的就是A4纸。
- **Height(纸高)** 键入 **H** 在弹出式样窗口中修改打印输出时所使用的纸张高度。省略时系统将其设置为A4复印纸的高度。此项亦不必更改。
- **Magnification(放大)** 放大打印。省略时系统取标准值为1000, 改变相对数值, 将得到放大或缩小的结果。例如选择1200时, 将标度放大1.2。
- **X-offset(左端边距)** 此项用于调节版心与纸面左边的距离(一般不要轻易去修改此参数, 用户可在自己的源文件中通过订口或切口尺寸的控制命令来调节)。省略时系统设置的左边距为1英寸。
- **Y-offset(顶端边距)** 此项用于调节版心与纸张顶部的边距(一般不要轻易去修改此参数, 用户可在自己的源文件中通过天头尺寸的控制命令来调节)。省略时系统设置的顶端纸边距为1英寸。
- **Batch(批处理模式)** 此选择项只有打开(On)和关闭(Off)两种状态(可通过上移键或下移键加回车键来选择)。关闭时系统在给出警告信息不会因等待用户键认面中键暂停。省略时处关闭状态。
- **Reverse order(反向检测开关)** 此选择项专供Canon激光打印机反向信号检测使用。使用方法同上。
- **Landscape(版面打印方式)** 此项用于选择在文件输出时版心纵向是否与长方形纸面的竖直方向相同。省略时系统取两者相同方式。
- **Input File(输入文件名)** 需要打印的DVI文件名。扩展名[.DVI]可以不键入。
- **Output file(输出文件名)** 可以用激光打印机输出的HP数据文件名。扩展名[.HP]可以不键入。
- **Log File(LOG 文件名)** 需要打印的LOG文件名。扩展名[.LOG]可以不输入。
- **Font sub(替代字库文件名)** 用于输入新的字库文件以替代系统缺少的字库。
- **Option File(系统排版命令文件名)** 用于读入存贮于系统中的可执行 $\text{\LaTeX}$ 排版命令。
- **Select TeX pages(选择输出页面)** 用于确定打印键出的DVI文件的页码数的范围。当文件校样稿很长, 页数较多时可控制系统只输出文件的某些页面。此项可

省略, 省略时系统自动为你将打印驱动程序设置为DVI 文件全部输出状态。输出页码范围的输入方式如下面的输出页码范围 项所述。

- #Pagelist(输出页码范围) 此选择项的功能同上, 主要用于确定打印输出的DVI 文件的页码数的范围。输出页码范围的控制方法如下:

- 对于单页输出, 只要在输入对应的页码数后按回车键即可;
- 需要打印某个页码区域时, 可输入该区域对应的上下限页码, 并在两个页码数之间插入一个冒号“:”。如果省略下限或上限, 则系统会把下限取为DVI 文件的最前面一页或将上限视作最后一面。例如“1:4”表示打印第1、2、3、4 页, 而“:5”则表示打印第五页之前(包括第五页)的所有页;
- 若要打印若干个页面区间时, 可根据上述规则定义的打印区域用分号“;” 隔开后按顺序输入。例如

:3;5:6;8;10:

表示打印1,2,3,5,6,8 页和第10 页开始的所有页。

- Pixel paths (examine)(选择打印字库路径) 系统已为用户设置好L<sup>A</sup>T<sub>E</sub>X 字库数据文件的路径(\pctex\pixel)。除非用户自己在另外路径配备了特殊字库数据文件, 一般不要去修改此项参数。
- TFM paths (examine)(选择排版字库路径) 系统在排版输出时, 可在TFM (T<sub>E</sub>X font Metric) 文件中得到各种字体或字号的有关信息。如果用户自己没有在其它路径配备了特殊字库的TFM 文件, 就不必去修改此项参数。省略时系统设置的路径为“\pctex\textfms”。

当选择了DVI 文件名以及相关的参数后, 按下回车键即可运行驱动程序PTIHP.EXE。运行任务结束后, 在磁盘的当前目录中会生成一个扩展名为.hp 的数据文件。需要硬拷贝时, 可再运行激光打印驱动程序PTISPOOL.EXE, 具体指令为:

C:>ptispool 文件引用名.hp -p=lpt1

## 3

# 基本行排命令

### 3.1 转码控制符

L<sup>A</sup>T<sub>E</sub>X 排版系统为用户提供了100多个控制命令。每个控制命令可以实现一定的排版功能。所有的命令都是以反斜杠“\”开头，在L<sup>A</sup>T<sub>E</sub>X排版程序语言中，反斜杠“\”被称之为转码控制符。当用户在进行编辑的小样文件(源程序)中的某个位置输入这个控制字符时，实际上就是在命令L<sup>A</sup>T<sub>E</sub>X暂时从前面的某种文字排版状态下退出，以接受新的排版命令，即要让系统知道对后面的文稿应作何种特殊的排版处理。

紧接着反斜杠转码控制符，通常是若干个字符组成的命令串。其目的是告诉L<sup>A</sup>T<sub>E</sub>X想要它为用户干什么。在L<sup>A</sup>T<sub>E</sub>X语言中，这种说明词类型的命令字符串与前面的反斜杠一起构成了一个标准的排版控制命令。在转码控制符“\”后面的命令串中，字符少的仅有一个，多则可能有二十来个。若按字符类型分类的话，归纳起来控制命令主要有两大类：控制字符类命令和控制词类命令。

#### 3.1.1 控制字符类命令

控制命令仅由反斜杠转码控制符和一个(只有一个)特殊的ASCII码字符构成，其中一些还须带参量。这类控制命令在L<sup>A</sup>T<sub>E</sub>X排版程序设计中虽然字符很少，却能完成许多特殊的排版功能。下面列举了其中的一些排版功能。

① 打印特殊控制字符 例如控制命令“\\$”可使L<sup>A</sup>T<sub>E</sub>X系统为你打印出美元记号\$，而“\%”和“\s”则可分别输出一个百分号与章节号“§”。

② 给字母顶上加特殊记号(亦称戴帽) 例如控制命令“\^{}{U}”可使L<sup>A</sup>T<sub>E</sub>X系统为字母“U”带上尖角帽而打印成“Û”。又如“\v{e}”和“\’{e}”则可分别输出“ë”和“é”。

③ 去除字母某个部分 例如控制命令“\i”和“\j”可使L<sup>A</sup>T<sub>E</sub>X系统为你把两字母“i”和“j”中的圆点去掉，从而分别打印成“i”和“j”。

④ 调节特定位置上的两字符间距 例如控制符“\,”可令L<sup>A</sup>T<sub>E</sub>X为你在两个字母间留出大约四分之一字母宽度的较小距离，而控制命令“\!”可使后面的字符倒退四分之一的字母宽度。

⑤ 换行排版 行文中插入双反斜杠转码控制符“\\”，可使排版输出时达到强迫换行的作用。在排标题时常常会用到这个控制命令。

⑥ 转换排版模式 我们知道科技论文正文中一般是由文字和数学公式构成的。L<sup>A</sup>T<sub>E</sub>X 系统将这两类内容分别按行文和数学模式进行排版。在这两种模式中的切换方式之一通过“\[” (进入独立数学模式) 和“\]” (退出独立数学模式) 来实现的。

### 3.1.2 控制词类命令

控制命令说明词是由若干个英文大、小写字母构成：有的直接取自英文单词(例如 \large、\LARGE)，有的是简写形式(如 \rm (罗马字体或白正体)、\bf(黑体)) 有的还必须要带上参变量(如 \documentstyle[12pt]{book})，有的则要与另外的一个控制命令说明词配对使用(如 \begin{ 参量 }, ..., \end{ 参量 })。所有这些说明词的可读性非常强。在这种类型的说明词构成的排版控制命令后面出现的任何一个非字母类 ASCII 码字符将自动给出命令结尾信息。例如阿拉伯数字、标点符号、空格等。许多使用 L<sup>A</sup>T<sub>E</sub>X 进行小样文件编辑的工作者经常喜欢采用一个或多个空格(空格多少无所谓，多余的会被系统忽略)记号作为控制命令自然结尾的记号，这样做既能防止命令与正文混淆而产生的错误，又能使小样文稿结构清楚，可读性强。但是要注意此时空格字符本身的空格排版功能失效了。假如排版命令前面的正文与命令后面的内容之间必须要空出一点距离的话，则不能再使用空格字符。此时比较好的方法是采用一对花括号“{}” (对外界排版环境无影响) 作为命令的结尾记号。例如我们在第二章排版实例2 中的打印出 L<sup>A</sup>T<sub>E</sub>X 专用记号的控制命令“\LaTeX{}” 尾端就采用了“{}”。这里读者会得到启示，带参变量的作版控制命令不用考虑结尾控制符的问题。在本章3.4 和3.5 节中我们还会就花括号“{}” 作进一步讨论。

作为控制词类排版控制命令应用实例，我们介绍一个打印英文日期的控制命令“\today”(今天) 的使用方法。“\today” 的作用是在 L<sup>A</sup>T<sub>E</sub>X 排版至该命令出现的地方时自动插入执行排版时的日期<sup>1</sup>。例如在小样文件中输入

```
''This document printed \today''
```

则在计算机内部日期为1992 年9 月1 日时会打印出

```
"This document printed November 1, 1992"
```

注意在上面控制命令“\today” 后面的第一个单下引号起到了终止命令的作用。但是假如我们想打印出“November 1, 1992 is a nice day.”，则决不能只是简单输入

```
''\today is a nice day.''
```

<sup>1</sup>一般中、低档计算机(如 IBM PC XT 等) 开机启动 MS-DOS 系统时需要事先设定正确的日期和时间。倘如未予设置，则日期出错。此时可通过运行 MS-DOS 系统内部日期更改命令 DATE 来更改。对于高档计算机(如 AT386、486 等) 则无须担心此事。



因为正如前面已经指出的，在“\today”后面的空格只是起到了终止命令的作用，而原有的空格功能已不再具备。所以你将看到的只会是“1992”与英文单词“is”连在一起的打印结果

“November 1, 1992is a nice day.”

解决这个问题现在已没有困难，读者只要在命令“\today”后面再添上一对花括号“{}”就可以了，即将小样文件修改为

“\today{} is a nice day”

就能输出你所希望的样式了。

用英文字母组成的字符串与转码控制符构成的一般排版控制命令在 $\text{\LaTeX}$ 排版程序设计中使用相当方便，读者会在后面的章节中体会到这一点。

## 3.2 字体转换命令

$\text{\LaTeX}$ 为用户提供了多达256种不同的西文字体。但是在系统中最常用的西文字体主要有七种：罗马字体(国内印刷业亦称之为白正体)、意大利字体(斜体)、黑体、倾斜字体、等线体(无衬线字体)、小体大写字母和打印机字体等。所有字体的转换命令形式相似，且非常简单。除了必加的转码控制符之外，字体说明词均由两个字母(英文字体的缩写)组成。例如在正文某个位置起希望将后面的字体用黑体打印出，则程序编辑者可在该处插入控制命令“\bf”。只要没有新的字体转换命令出现，原来的字体形式会在行排中一直延续。常用的字体、控制命令以及打印机输出的效果加表3.1所示。

表 3.1 西文字体转换命令表

西文字体	控制命令	源程序实例	排版效果
罗马字体	\rm	\rm Roman type	Roman type
意大利字体	\it	\it Italic type	<i>Italic type</i>
黑体	\bf	\bf bold face type	<b>bold face type</b>
倾斜体	\sl	\sl slanted type	<i>slanted type</i>
等线体	\sf	\sf sans serif type	sans serif type
小体大写字母	\sc	\sc small caps type	SMALL CAPS TYPE
打印机字体	\tt	\tt typewriter type	typewriter type
数学斜体	\mit	$\mit Italic math type$	<i>Italic math type</i>

表3.1中列举的罗马字体是最常用的印刷字体。在 $\text{\LaTeX}$ 小样文件中，如果对字体未作特别注解，则小样文件中的英文或其它西文均以罗马字体排版。意大利字体是与罗马字体风格迥然不同的印刷字体，常被用来编排文件中需要强调的地方。黑体则被

用来编排章节标题和文章中应特别强调或突出的地方。倾斜体有点像意大利字体，主要区别是在倾斜体是将罗马字体简单向右作适当倾斜(罗马字体的底部不动)。注意在使用意大利字体或倾斜体结束后转换至其它正体字时不要忘记补加进一些额外的空距。因为按照规则，字与字之间是根据一条隐含的接近每个字底部的基准线对齐并按照一定的字距来排版的。由于斜体字上端相对底部向右作适当位移，所以在与其它的正文合排时斜体字的上端会与邻近的正体字靠得太紧，尽管两字间的底部隔开了一点距离。比较简单的添加额外距离的方法是在斜体与正体之间插入控制命令“\/”。例如小样“\it He did\rm n't.”会排成“*He didn't.*”。假如在两种字体转换的位置插入控制符“\/”，即输入排版程序

```
\it He did\/\rm n't.
```

则可获得较为满意的排版结果“*He didn't.*”。

等线体字常被用于制作幻灯片或文章的醒目标题。至于打印机字体，实际上是相对排字而言的。这是因为无论你选择一个什么样的非电脑类(如机械式或电动式)的打字机，用其打印出来的文件拷贝与一般出版商印刷出来的文件拷贝总有一些本质上的差别。首先，每个字符出现在纸上时总要占据固定不变的宽度，而印刷物上的铅字却不是这样，每个字符出现在纸上时所占据的宽度是与字本身宽度有关的。例如对于打印机字体字母“i”和“w”在纸上占据的宽度是相同的，但是对于实际印刷刊物中出现的字体，“i”比“w”要窄得多。再者，西文印刷物中各个铅字之间还会根据字形特征作相对位移。例如“V”和“A”在排版时“V”字的右边界会与“A”字的左边界靠拢交叠而排成“VA”。此外，印刷物中一些特定的字符串还会以专用联体字形式排出。例如“f”和“i”合在一起时成为“fi”，而“f”、“f”和“f”在一起会排成“fff”。

在 $\text{\LaTeX}$ 系统中还有一种字体形式不确定的特殊字体：强调(emphasis)字体。在执行排版时，根据行文中原来所选取的字体，系统会自动将强调字体在意大利字体与罗马字体之间切换。强调字体控制命令是“\em”，它的功能是在用户需要强调的文章中某个部分用醒目的字体印出。字体的转换规则是：当正文排版至强调字体转换命令处时，如果当前行文是罗马字体的话，后面内容则将以意大利字体输出；假若行文原来选取的是意大利字体，则系统又会将字体变换到罗马字体。在使用此命令打印完所要强调的内容后不要忘记再使用字体转换命令切换到原来的正文字体状态下。

表3.1中最后一行列举的数学斜体字主要用于编排数学公式，这种字体必须是在特定的数学公式环境中方能使用(例如表3.1中的程序实例就是通过控制将 $\$$ 来设置和退出数学公式环境的)。除了常规的英文数学斜体字体外， $\text{\LaTeX}$ 另外还提供了一些在数学公式中常用到的特殊字体，如希腊字母和书写体。这些字体同样需要在特定的数学公式环境中才能调用，此外有些字体的每个字母都要有专门的打印控制命令。我们将在第六章中详细介绍这方面的知识。读者若需要可先查阅第六章的有关内容。

上面介绍的字体转换命令虽然比较简单，但是没有确定字体命令的有效范围。如果需要对字体转换命令作用区域加以说明，则可采用以下等价的配对控制命令：

```
\begin{字体说明词}
```

(西文正文)

\end{字体说明词}

字体说明词参量可选用以上由两个字母构成的字体单词简写式, 如“rm, it, em”等。例如下面的排版源程序

```
Words are separated by \begin{em} one or more spaces. \end{em}
Paragraphs are separated by \em one or more blank lines. \rm
The output is \sf not affected by adding extra spaces \rm
or \begin{sf} extra blank lines \end{sf} to the input file.
```

```
Double quotes are typed like this: '\begin{tt}quoted text\end{tt}'.
Single quotes are typed like this: '\tt single-quoted text \rm '.
```

```
\sl Long dashes \rm are typed as
\begin{sl} three dash characters\end{sl}---like this.
```

```
Italic text is typed like this: \it this is italic text.\rm
\bf Bold text is typed like this: \bf this is bold text.\rm
```

可以输出

Words are separated by *one or more spaces*. Paragraphs are separated by *one or more blank lines*. The output is not affected by adding extra spaces or extra blank lines to the input file.

Double quotes are typed like this: “quoted text”. Single quotes are typed like this: ‘single-quoted text’.

Long dashes are typed as three dash characters—like this.

Italic text is typed like this: *this is italic text*. Bold text is typed like this: **this is bold text**.

### 3.3 字号转换命令

在印刷业中, 字号是表示排版用的铅字面积大小的一个术语。在汉字排版时, 计算铅字面积大小通常采用号数制和点数制两种方法。号数制是中国国内用来计算汉字铅活字大小的标准制度。除了特殊美术字体外, 对于一般的正方形汉字, 号数直接反映了铅字的高或宽的线度大小。例如一般书刊的标准字体为5号字(本书亦采用5号字)。点数制是国际上流行的计算西文铅活字大小的标准制度。点数制的基本计量单位“点”(亦称“磅”)是由英文“Point”翻译的, 缩写为P或pt。由于西文字母(例如英文, 法文等)各字身的宽度不尽相同, 因此, 西文字体的点数是按基本字身的高度(长度)来计量的。在点数制中, 规定一英寸有72.27点, 一点等于0.3515毫米。

LaTeX 各种字号的命名并不是像中文字按数字号码来区分的, 此外每一种字号的大小也并不是绝对的。字号的意义完全是相对正文基准字体的大小而言的。所以实际打印出来的某一种字号的大小取决于你在小样文件第一行里设置的版心正文基准字体的尺寸。LaTeX 系统有三种基准字号(10pt, 11pt 和 12pt) 可供选择。如果你未在源程序中予以设置(省略), 那么 LaTeX 就会将正文基准字体自动设置为 10 点(接近中文印刷体的五号铅字, 本书的正文部分就是用约 10 点宽度的中文字体排版的, 约等于 3.5 毫米宽)。如果你设置了 11 点(pt), 那么所有各号字体的线度会相应地放大百分之十。LaTeX 系统总共定义了十大类字号的各种字体。这些字号的转换命令是通过描述尺寸大小写的英文单词构成的。在表 3.2 中我们列出了可使用的十类字号的控制命令以及打印的结果(注意表中的版心基准字体设置为 10 点)。

表 3.2 字号转换命令表

点数(pt)	相应中文字号	控制命令	实际打印效果
25	一号	<code>\Huge</code>	Tongji
20	二号	<code>\huge</code>	Tongji
17	三号	<code>\LARGE</code>	Tongji
14	四号	<code>\Large</code>	Tongji
12	小四号	<code>\large</code>	Tongji
10	五号	<code>\normalsize</code>	Tongji
9	小五号	<code>\small</code>	Tongji
8	六号	<code>\footnotesize</code>	Tongji
7	小六号	<code>\scriptsize</code>	Tongji
5	七号	<code>\tiny</code>	Tongji

说明:

① 使用各个控制命令时要注意每个英文字母大小写的区别。将上述各类字号在实际输出时的大小与一般西文印刷刊物比较, 你会发觉命令“`\Huge`”和“`\huge`”可用来排大标题, 而控制命令“`\LARGE`”、“`\Large`”和“`\large`”则可排各章节的小标题; 正文部分的主要字体(主字)编排可采用“`\normalsize`”来控制, 亦可省略; “`\small`”常用于编排图表中的注文; 命令“`\footnotesize`”可用于脚注或篇后注; 而命令“`\scriptsize`”和“`\tiny`”则用作编排数学公式中的上下标等。

② LaTeX 在接受表中的任何一个命令后, 立即会无条件地将此命令后面的文字放大或缩小打印。不过要提醒读者的是后面输出的文字在改变尺寸后又返回到版心所用的罗马字体。所以在行文中插入这些字号转换控制命令以后, 你还须明确一下后面正文所需的字体。例如你想打印一行比基准字体(10pt) 小一号意大利字体的英文字

“*small italic commentary*”, 那么你应该在源文件中输入

```
\small \it small italic commentary
```

如果将意大利字体控制命令“\it”放在控制命令“\small”之前, 那么字体转换令无效, 输出的将仍是罗马字体。

③ 表3.2 中列出的字体大小的打印样式只是在基准字体为10pt 的情形下获得的。对于基准字体设置为11pt 或12pt 的场合, 各种字号转换命令控制打印出的西文字体的实际尺寸如表3.3 所示。

表 3.3 不同基准字体状态下字号转换命令控制的实际尺寸

字号转换命令	10pt 基准字体	11pt 基准字体	12pt 基准字体
\tiny	5pt	6pt	6pt
\scriptsize	7pt	8pt	8pt
\normalsize	10pt	11pt	12pt
\large	12pt	12pt	14pt
\Large	14pt	14pt	17pt
\LARGE	17pt	17pt	20pt
\huge	20pt	20pt	25pt
\Huge	25pt	25pt	25pt

④ 在用户所使用的 $\text{\LaTeX}$  系统中, 每种字号对应的各种字体并不一定会全部被安装了进去。这时系统一般会调用同样字号的罗马字体来取代, 当然你也可在系统给你警告提示缺少某种特殊字体的情况下, 在源文件中对字体转换命令作些修改, 从而用其它替代字体来输出你的文件。也许你的系统另外安装了一些特殊字库, 你可以查阅系统配备的操作指南(Local Guide), 以了解具体的使用方法。表3.4 列出了2.09 版本的 $\text{\LaTeX}$  排版系统中可供使用的每种字号所对应的各类有效西文字体。

其中“P”表示可由字号与字体转换命令直接调用的有效字体(这种类型的字体在启动 $\text{\LaTeX}$  时就被装入基本内存)。“D”表示需要时必须先使用调字库命令

```
\load{ 字号转换命令}{ 字体转换命令}
```

将字体装入系统, 然后才能使用。例如从表3.4 中可知, 需要8pt 字号(若基准字体为10pt 时, 则由命令“\footnotesize”控制) 所对应的打字机字体(由命令“\tt”控制) 时, 可使用如下命令:

```
\load{\footnotesize}{\tt}
\footnotesize \tt
.....
```

表3.4 中的“X” 表示用户拥有的 $\text{\LaTeX}$  排版系统中没有此种字号的西文字体。例如用户不可能在2.09 版本的 $\text{\LaTeX}$  系统中打印出5pt 大小的等线体西文字母。

表 3.4 各种字号的有效字体详表

字号	\it	\bf	\sl	\sf	\sc	\tt
5pt	D	D	X	X	X	X
6pt	X	D	X	X	X	X
7pt	P	D	X	X	X	X
8pt	P	D	D	D	D	D
9pt	P	P	D	D	D	P
10pt	P	P	P	P	D	P
11pt	P	P	P	P	D	P
12pt	P	P	P	P	D	P
14pt	D	P	D	D	D	D
17pt	D	P	D	D	D	D
20pt	D	D	D	D	D	D
25pt	X	D	X	X	X	X

### 3.4 分组控制符——花括号用途之一

在本章的前两节里，我们介绍了各种字体和字号的转换命令。初学者在使用这些命令时常犯的错误是在对某些文字作字体变换后会忘记再将原来所用的字体恢复。L<sup>A</sup>T<sub>E</sub>X 为解决这个问题专门设置了一个用一对花括号“{}”构成的分组控制符。它的作用是在小样文件中设置一个由花括号作为前后边界的特殊排版群体的小组环境。在这小环境中的任何字体、字号或其它排版令导致的变化对外界无丝毫影响。例如我们想要打印“The bold Roman marched home”，仅用第二节学到的知识，我们可输入

```
The \bf bold \rm Roman marched home
```

但是我们现在还可使用另外一种更简单的方法即用分组控制符来设置打印句中黑体的小环境：

```
The {\bf bold} Roman marched home
```

花括号分组控制符可以连续嵌套使用，以设置不同层次的排版小环境，灵活使用分组控制符可使你的排版程序简单且不易出错。当然不要在嵌套使用分组控制符时缺少了配对花括号的某一部分(左花括号或右花括号)。

在上面读者看到了继反斜杠转码符后又一对L<sup>A</sup>T<sub>E</sub>X的控制字符的重要作用。假如读者想要在行文中打印出花括号的话，千万不要忘记在左或右花括号前加上一个反斜杠字符。否则，在排版时L<sup>A</sup>T<sub>E</sub>X只会将其视作控制字符处理。

## 3.5 参变量控制符——花括号用途之二

花括号除了在 $\text{\LaTeX}$ 系统中充当分组控制符之外, 它还具有类似带变量数学函数中的括号功能, 为许多 $\text{\LaTeX}$ 排版控制命令添置数学变量或补充说明设置参变量环境。例如在第二章中介绍的文件版式命令“ $\text{\documentstyle}$ ”后面必须要带有一个说明文体类型的参量, 假如是考虑编排报告类文件, 则应将命令添加注解补充成

```
\documentstyle{report}
```

作为在花括号内注释用的参变量主要有两大类。一类纯粹是注解说明式的, 其功能只是对控制命令方式的补充, 输出时无实质内容打印出来; 另一类则不只是对控制命令方式的简单补充, 它自身又构成一个字、词、或段落的打印实体(在电子排版系统中通常称之为盒子), 并且允许在花括号内对这些打印实体继续嵌套插入排版控制命令。

当然, 花括号不是唯一用作为排版命令设置参变量的控制符, 就像前面我们已经遇到的, 方括号偶尔也可在一些控制命令中为添置参变量设置环境。但是要注意的是与花括号有本质上的差别, 方括号不能单独成为控制字符。因为方括号在非控制令行文中出现时其作用与其它字母一样, 只是在硬拷贝中输出方括号字符。所以不要随意将标准控制命令中的标明参变量用的花括号用方括号来取代, 这会导致严重的排版错误。

## 3.6 尺寸控制命令

有时用户想要在某一页的特定位置留出些空白来编排图表, 这时用户必须得告诉 $\text{\LaTeX}$ 系统两件事情: 第一, 在版面上留出空白间距的方向(是纵向还是横向); 第二, 所留空距的大小和计量单位。

### 3.6.1 $\text{\LaTeX}$ 长度计量单位

$\text{\LaTeX}$ 排版系统的长度计量单位有许多种, 如前面多次提到的打印机的“点(pt)” (1点 = 1/72.27 英寸)、英寸(in)、厘米(cm)、毫米(mm)等。表3.5列出了一些常用的计量单位。在编辑排版源程序过程中, 用户可以根据自己的需要和习惯来选择使用不同的单位。

与其它计算机排版软件相比,  $\text{\LaTeX}$ 在空白间距的连续可调方面可达到极高的精度, 并且使用非常方便。长度单位通常是以参量的形式被一些带数值参量的间距控制命令所调用(参见表3.5), 其大小数值不仅允许为小数, 而且还可取负数。

有时, 更方便的方法是使用基于当前选用的基准字号的相对尺寸单位。这种单位有两个: “em”和“ex”。单位em用来表示水平间距, 其大小相当于大写字母“M”的宽度, 而单位ex则被用来表示垂直(纵向)间距, 其大小相当于小写字母“x”的高度。

表 3.5 常用的 L<sup>A</sup>T<sub>E</sub>X 长度计量单位表

计量单位	意义	实际长度(毫米)	控制命令实例
pt	点(磅)	0.35	<code>\headsep=25pt</code>
in	英寸	25.40	<code>\vspace{1.5in}</code>
cm	厘米	10.00	<code>\textheight=21.5cm</code>
mm	毫米	1.00	<code>\textwidth=155 mm</code>

### 3.6.2 L<sup>A</sup>T<sub>E</sub>X 横向空白间距控制命令

L<sup>A</sup>T<sub>E</sub>X 排版系统为用户提供了使用极其方便的横向空白间距控制命令。它是以格式为

`\hspace{横向空白间距}`

的排版命令来实现的。其中空白间距的单位可选用表 3.5 中的各项。例如：

```
Here\hspace{1.0cm}is a 1 cm space.\\
Here \hspace{1.0cm}is a 1 cm space including a blank space.\\
Here \hspace{1.0cm} is a 1 cm space including two blank space.\\
Negative space is a backspace---like this.\hspace{-0.5in}/////
```

可以获得

```
Here      is a 1 cm space.
Here      is a 1 cm space including a blank space.
Here      is a 1 cm space including two blank space.
Negative space is a backspace---like this.
```

从上述例子说明在行文中的某一行内插入带长度参量的空距命令“`\hspace`”可使命令所在位置空出用户所设定的空白距离。但是要注意，在段首或段尾处此命令无效。如果用户必须要令系统为你在上述控制命令无效的某个部位插入横向空距，那么还可使用带星号“\*”的无条件横向空距命令

`\hspace*{横向空白间距}`

其功能与前一个命令相同，只是不会再为用户考虑版面上是不是该加空距。

### 3.6.3 L<sup>A</sup>T<sub>E</sub>X 纵向空白间距控制命令

#### 一、带行间距参数的换行控制符

在第 2.3.2 小节中，我们介绍了双反斜杠“`\\`”换行控制符。如果需要在强迫换行的当前行与后续行之间另外再插入一些空白行间距，那么还可以在控制符“`\\`”后面再添



加一个用方括号设置的长度参数:

`\[ 行间距]`

其中行间距 可以为带 $\text{\LaTeX}$  长度单位的正实数。这样, 用户就可以自行调节当前行与后续行之间的间距了。

注意, 此命令有一个带星号“\*” 的特殊表达式:

`\*[ 行间距]`

与无星号控制命令的差别仅在于有星号的换行控制命令不会支持在换行位置断页(参见 10.11)。

## 二、可调纵向空白间距控制命令

$\text{\LaTeX}$ 系统中提供的可调纵向空白间距命令是用来控制行文排版过程中在垂直方向上需要留出的空白大小。例如在段与段之间、文中插图或表格等需要增加空距的地方就需要用到此命令。它的标准格式为

`\vspace{ 纵向空白间距}`

例如:

```
You\vspace{6.5mm} seldom add space like this
between lines in a paragraph, but you sometimes
remove space between them by adding some
negative space.
```

```
\vspace{7 mm}
```

```
You more often add space between
paragraphs---especially before or after displayed
material.
```

执行排版后可以获得

```
You seldom add space like this between lines in a paragraph, but you sometimes
remove space between them by adding some negative space.
```

```
You more often add space between paragraphs---especially before or after dis-
played material.
```

如果此命令恰好落在某一页版面的顶部, 则命令无效。假如用户必须要令系统为你在此空出一些间距, 那么还可使用带星号“\*”形式的无条件纵向空白间距命令

`\vspace*{纵向空白间距}`

其功能与上面命令相同, 只是不会再为用户考虑版面上是不是该加空距。

### 三、纵向固定空白间距控制命令

除了上面介绍的用具体尺寸来调节段与段或行与行之间的空白间距之外, 还可使用以下三种固定尺寸(大、中、小)的纵向空白间距控制命令:

`\smallskip`

`\medskip`

`\bigskip`

例如程序

```
Here is an example of \verb|\smallskip| command.\\
\smallskip
This is an example of \verb|\medskip| command.\\
\medskip
And what you see now is\\
\bigskip
the example of \verb|\bigskip| command.
```

可以获得三种不同尺寸的行间距拷贝:

```
Here is an example of \smallskip command.
This is an example of \medskip command.
And what you see now is
the example of \bigskip command.
```

以上介绍的三个不带参变量的控制命令有完全等价的控制命令

```
\vspace{\smallskipamount}
\vspace{\medskipamount}
\vspace{\bigskipamount}
```

而其中的三个长度参量

`\smallskipamount`    `\medskipamount`    `\bigskipamount`

的数值(行距大小)由具体的文件版式确定。

### 3.6.4 长度控制命令的赋值命令

L<sup>A</sup>T<sub>E</sub>X 排版系统中有许多尺寸控制命令本身就已赋予了具体的长度数值(在源文件全局排版说明区域内定义或按省略规则确定)。例如版心宽度命令“\textwidth”和版心高度命令“\textheight”等。如果用户并不准备改变原来命令中所隐含的长度数值, 在使用这些长度控制命令时, 就不需要再作尺寸大小的任何说明。假如用户在一些特殊版面上希望改变某个长度控制命令原来所赋予的数值, 则可采用类似通常计算程序语言中规范的赋值方法:

$$\langle \text{长度控制命令} \rangle = \langle \text{长度数值} \rangle$$

例如:

```
\textwidth= 14.5 cm
\textheight= 21.5 cm
```

在L<sup>A</sup>T<sub>E</sub>X 排版系统中, 另外还为用户提供了一些专门的赋值控制命令。与上面命令完全等价的赋值控制命令为:

```
\setlength{长度控制命令}{长度数值}
```

用此命令可将前面的排版程序实例改成:

```
\setlength{\textwidth}{14.5 cm}
\setlength{\textheight}{21.5 cm}
```

如果用户并不详细知道原来的长度控制命令中具体的长度数值, 只是希望在原来的尺度上增加一点数值, 那么还可采用如下添加数值的控制命令:

```
\addtolength{长度控制命令}{长度增加值}
```

例如当用户准备将原来的版面拓宽1cm 时, 可输入如下控制命令:

```
\addtolength{\textwidth}{1 cm}
```

如果用户希望控制的尺寸只是间接数值(例如相当于某段文字的长度) 那么不必去测量长度的具体细节, 可直接使用以下命令:

```
\settowidth{长度控制命令}{字符串}
```

其中的字符串 参量必须是LR 排版盒子(参见3.7 节) 状态下的单行短文。排版时, 系统会根据字符串 实际内容虚拟排印, 然后将获得的长度再对长度控制命令 赋值。

## 3.7 LR 排版盒子

L<sup>A</sup>T<sub>E</sub>X 系统排版状态主要有行文排版状态、数学公式排版状态、LR 排版状态 和

图形排版状态等。一般的行文排版状态就是本书到此页为止介绍的各种排版程序实例中所使用的排版环境(源文件中只要没有特别加以排版状态的说明,那么系统总是处在行文排版状态下)。它的特点是排版时除了能将行文自左向右排版外,还能自动往纵向方向逐行排版;另外,还能实行自动翻页排版等。数学公式排版状态是专为编排数学方程式而设计的排版模式(关于数学公式排版状态的设置可参见第六章),在这种排版状态中,普通的行文排版的许多规则不能简单使用。

LR 排版状态(亦称单行行文排版状态)是行文排版状态的特例。它只能实现从左到右的文字编排工作,而不能向纵向发展。通常被用在数学公式排版状态和图形排版状态中等需要插入不能拆开的文字整体的场合。

LR 排版状态可以通过建立LR 排版盒子来实现状态切换。“排版盒子”是 $\text{\LaTeX}$ 排版系统中一个非常重要的概念。所谓排版盒子(box)是指排版过程中的一个独立的矩形排版单位,排版系统会将盒子视作一个不可拆开的整体。无论在换行或翻页时,盒子内的内容不会被分离排版。此外,盒子如同一个巨型西文单词,在与其它文字混排时,盒子与单词间的空格或空行功能与一般行文中的出现的空格或空行功能完全相同。建立LR 排版盒子可使用下面介绍的无框线LR 排版盒子或带框线LR 排版盒子控制命令。

### 3.7.1 无框线LR 排版盒子

无边界线框LR 排版盒子(makebox)的控制命令为:

`\makebox[盒子长度][对位参量]{单行短文}`

其中盒子长度的单位可采用在本章第六小节中介绍的 $\text{\LaTeX}$ 排版系统标准的各种长度计量单位。命令中第二个方括号内的对位参量是用来对盒子内的文字排版方式作具体位置注解。这部分注解内容可以省略。省略时系统会将盒子内的文字内容在盒子小环境内居中排版。如果你不喜欢居中排版方式,那么还可选取另外两个排版方式。根据需要,你可在方括号内填入下面两个参量中的一个:

- ① l 盒子内的文字紧靠虚拟盒子的左边界排版;
- ② r 盒子内的文字紧靠虚拟盒子的右边界排版。

例如:

```
The word \makebox[3cm]{\em center} is here.\\
The word \makebox[3cm][l]{\em left} is here.\\
The word \makebox[3cm][r]{\em right} is here.
```

输出结果为:

```
The word      center      is here.
The word left              is here.
The word              right is here.
```

命令“\makebox”有一个功能相似的缩写式:

`\mbox{单行短文}`

注意无线框LR 排版盒子命令缩写式“`\mbox`”没有盒子长度参量和对位参量可供选择。实际排版过程中盒子的长度将根据盒子内的文字多少来自动调节。

### 3.7.2 带框线LR 排版盒子

带框线LR 排版盒子的控制命令是

`\framebox[盒子长度][对位参量]{单行短文}`

命令“`\framebox`”与“`\makebox`”功能完全相同，只是再用醒目的边框线将矩形盒子围起来而已。这个命令同样配备了一个功能类似的但没有盒子长度和对位参量可供选择的缩写形式：

`\fbox{单行短文}`

实际排版过程中盒子线框的长度将取决于盒子内的文字多少。例如：

The `\verb|\framebox|` command puts a `\framebox[3cm][l]{\em frame}` around the outside of the `\fbox{\sf box}`.

输出结果：

The `\framebox` command puts a `frame` around the outside of the `box`.

注意：如果用户采用具有盒子长度和对位参量可供选择的盒子控制命令时，系统将严格按命令中定义的盒子长度将盒子作为一个整体来排版。如果盒子内部的文字内容排不下而被挤出盒子的左边或右边时，则命与邻近的字符发生重叠。例如小样：

X X X X`\framebox[1.2cm]{Superposition}`X X X X

命打印出

X X X`Superposit`~~ion~~X X X

当然，这种排版特性并不是排版软件的缺陷。系统在许多排版场合(如矩阵，表格等)，就是利用这一特性(使用长度为零的无框线盒子命令)实现纵向对位排版的。试看下面一段排版程序：

```
.....
\hspace*{4cm}\makebox[0pt]{Department of Physics}\\
\hspace*{4cm}\makebox[0pt]{Tongji University}\\
\hspace*{4cm}\makebox[0pt][l]{Department of Physics}\\
\hspace*{4cm}\makebox[0pt][l]{Tongji University}\\
\hspace*{4cm}\makebox[0pt][r]{Department of Physics}\\
```

```
\hspace*{4cm}\makebox[0pt][r]{Tongji University}\\
.....
```

输出结果:

```

:
:
Department of Physics
Tongji University
Department of Physics
Tongji University
Department of Physics
Tongji University
:
:
```

注意其中的第一、第二行即为居中排版, 第三、第四行为靠左边界对齐排版, 而第五、第六行则是靠右边界对齐排版。在数学公式排版环境中, 使用这种方法还可以非常容易地实现纵向各个数字小数点位置的对齐。

### 3.7.3 实心矩形盒子

实心矩形是指LR盒子内完全填充(涂黑)的矩形图, 一般用于在行文中描画较粗的尺子状线条或线框。实心矩形盒子的排版控制命令为:

```
\rule[盒子升高尺度]{矩形宽度}{矩形高度}
```

例如:

```
Rule one: \rule{2mm}{4.5mm} \hspace{1cm}
Rule two: \rule[3mm]{1cm}{1.5mm}
```

输出结果为:

```
Rule one:  Rule two: 
```

说明:

① 盒子升高尺度 是指盒子图形整体相对原基准线升高(或降低)实际长度。长度数值可正可负(负值表示图形降低的深度), 长度单位可任选一种 $\text{\LaTeX}$ 系统的合法单位。

② 矩形宽度和矩形高度不能取负值, 但可取0(仍须带长度单位)。当两者之中任意一个长度取值取0时, 直线已经看不到。但是另外一个配对的高度或宽度值在排版时仍然有效。利用这个性质, 在排版时, 可以用此命令调整版面, 例如取宽度数值为0时, 可调节盒子占版面的高度。试看下例:

```
Compare \fbox{\bf This box} with
\fbox{\rule[-0.5cm]{0cm}{1cm}\bf This box}
```

输出结果:

Compare This box with This box

③ 由于实心盒子内已经涂黑, 本命令不能用于在盒子内编排文字。

### 3.7.4 LR 排版盒子的版面控制

使用命令“\raisebox”可以设置一个类似“\makebox”命令所产生的LR 排版盒子, 并且能控制盒子在版面上相对原基准线升高或降低。命令格式为

`\raisebox{基线升高}[盒子高度][盒子深度]{盒子正文}`

说明:

① 盒子升高是指LR 盒子本身基线相对行文基准线升高(或降低)的实际长度。长度数值可正可负(负值表示LR 盒子降低), 长度单位可任选一种 $\text{\LaTeX}$ 系统的合法单位。

② 盒子高度和盒子深度分别指LR 盒子相对自身基线上方和下方延伸的空间尺度。省略时, 系统将根据盒子内的文字信息自动调整。例如:

Now you can `\raisebox{1ex}{\small \em raise}`  
or `\raisebox{-1ex}{\small \em lower}` text in the LR box.

输出结果:

Now you can *raise* or *lower* text in the form of LR box.

## 3.8 无框线LR 排版盒子的存取

使用无线框的LR 盒子排版命令, 或使用在图形状态下无线矩形作图命令(参见第8.3.2节)编制程序, 可以控制 $\text{\LaTeX}$ 系统为你编排文字或绘制含有文字信息的无线框矩形图形。然而在令 $\text{\LaTeX}$ 执行排版时, 需要一定的机时(CPU 时间)。特别是作图, 时间耗费尤其厉害。对于在一个输出文件中重复出现的长字符串, 或多次出现在同一张插图中的相同物体(含有文字信息的无线框矩形图形), 让系统根据源程序重复地计算编排行文和作图显然是一种没必要的浪费。 $\text{\LaTeX}$ 系统对于经常使用的无线框的LR 盒子排版命令和矩形作图命令专门设计了有关的排版信息存取控制命令。

本节着重介绍无线框LR 排版盒子内文字信息的存取方法。有关无线框矩形图形的存取方法将在第8.3.2节中再作介绍。

### 3.8.1 存贮库的命名

要在计算机内存里留出某个空间暂时存贮一行在无框线LR 排版盒子中的文字, 首先要给存贮库命名, 以使以后调用时有一个专用名。使用存贮库命名控制命令:

`\newsavebox{存贮库专用名}`

就可实现这种功能。其中存贮库专用名 必须是由反斜杠控制符“\”与合法的字符串构成, 例如你可定义一个专用名为“\toyNumone”的存贮库:

`\newsavebox{\toyNumone}`

注意不要采用 $\text{\LaTeX}$ 专用的排版控制命令来作为信息存贮库的名字, 也不要再次选用已经用过的存贮库名字。否则的话, 系统会拒绝执行排版, 并向你紧急报错。

### 3.8.2 无框线LR排版盒子的存贮

需要往已经命名的存贮库里存贮一个含有文字信息的无框线LR排版盒子, 可以使用命令:

`\savebox{存贮库专用名}[盒子长度参量][对位参量]{单行短文}`

这个控制命令中的选择参量与无边界线框LR排版盒子控制命令“\makebox”中的功能完全一样。执行排版时不会在版面上打印出任何拷贝, 它会将排版结果存贮到已经命名的存贮库里。以后调用时系统不再进行具体计算, 直接将结果当做一个“巨型字母”往版面上“贴”。例如你可通过下面一行小样在专用名为“\toy”的存贮库里存贮一个LR盒子:

`\savebox{\toy}[2cm][l]{Tongji}`

如同“\makebox”命令一样, 存贮LR盒子的控制命令“\savebox”亦有一个无选择参量的缩写形式:

`\sbox{存贮库专用名}{单行短文}`

它具有类似“\mbox”命令(见第3.7.1节)的控制功能, 两者的差别仅在于命令“\sbox”只会将排版结果送往存贮库而已。

为了节省内存, 用户可以多次使用同一个存贮库来存贮不同的文字信息。但是要注意的是, 你每次存入新的内容的时候, 系统会自动将原来存贮的东西全部清除。另外, 如果从某一版面起存贮库已经没有用处, 那么, 还可以采取存贮库“腾空”措施:

`\sbox{存贮库专用名}{}%`

### 3.8.3 存贮信息的调用

对于源程序中处在行文排版状态下的任意位置, 如果需要调用存贮库里的文字信息, 则可通过使用命令

`\usebox{存贮库专用名}`

即能达到迅速排版的效果。下面是一个存取信息的排版实例:



```

\newsavebox{\tju}
\savebox{\tju}[2.8 cm][r]{Tongji University}
\newsavebox{\phy}
\sbox{\phy}{Department of Physics}
.....
\begin{verse}
You are from the \usebox{\phy}, \usebox{\tju}. \\
He is from the \usebox{\phy}, \usebox{\tju}. \\
And I am from the \usebox{\phy}, \usebox{\tju} too.
\end{verse}

```

输出结果为:

```

You are from the Department of Physics, Tongji University.
He is from the Department of Physics, Tongji University.
And I am from the Department of Physics, Tongji University too.

```

### 3.9 西文特殊符号的排版控制命令

在编排西文文稿的过程中, 总免不了要输入一些特殊符号。例如前面一章介绍的被 $\text{\LaTeX}$ 作为专用排版控制符的符号(如计算机上键盘上出现的#, \$, % 等), 这些字符的打印并不难, 只要将反斜杠控制符“\”添加在相应的字符前面就可以了。但有些国家的文字(如法语, 德语和匈牙利语等)中会出现一些基本ASCII码中没有的符号, 例如英镑符号“£”、国际音标中的符号“æ”、匈牙利语中的字母“ö”等。在 $\text{\LaTeX}$ 行文排版状态或LR排版状态中, 这些特殊符号是通过专用的控制命令来产生的。表3.6中列出了六种特殊标识符号的排版控制命令。

表 3.6 特殊标识符号命令表

符号	控制命令	符号	控制命令	符号	控制命令
†	\dag	§	\S	©	\copyright
‡	\ddag	¶	\P	£	\pounds

例如在源文件中正文的任意位置插入命令“\copyright”便可获得国际上通用的版权标识符号“©”。以上六种符号的打印控制命令在其它排版状态中(例如数学公式排版状态)亦可使用。另外一些非英语类西文中常见的特殊符号(见表3.7)的排版控制命令就只能在行文排版状态或LR排版状态中使用了。

注意, 表3.7中最后一列中的两个颠倒的标点符号“¿”和“¡”是由原来的标点符号与单下引号(‘)配对控制产生的。如果在数学状态中使用表3.7中的特殊符号, 则必须先

表 3.7 西文(非英语类)特殊符号命令表

符号	控制命令	符号	控制命令	符号	控制命令	符号	控制命令
œ	\oe	ä	\aa	l	\l	¿	?'
Œ	\OE	Å	\AA	L	\L	¡	!'
æ	\ae	ø	\o	ß	\ss		
Æ	\AE	Ø	\O				

表 3.8 字母外形特征符号修饰控制命令表

字母	控制命令	字母	控制命令	字母	控制命令	字母	控制命令
ˆ	\^{v}	˜	\~{v}	˘	\v{v}	˙	\c{v}
˚	\' {v}	¯	\={v}	˚	\H{v}	˚	\d{v}
˘	\^{v}	˙	\. {v}	˚v	\t{vv}	˚	\b{v}
˚	\" {v}	˚	\u{v}				

在数学公式排版环境中用命令“\mbox{ }”建立一个LR排版盒子,然后再在花括号内填入表中的打印控制命令。

有些西文字母具有一些共同的特性:如法语中一些字母顶部带有一个撇标记,比如像字母“ê”,而在匈牙利语中会出现一些顶部带有两小点标记的字母。归结起来,不外乎是在英文字母身上作些特征修饰而已。所谓修饰,这里是泛指“带帽”、“穿鞋”、“连体”(如字符“Æ”)等。表3.8列出了对英文字母作常见的外形特征修饰的控制符命令。表中以字母“v”作为例子。用户可以采用相同的控制符对其它英文字母作外形特征修饰。

注意表3.8中介绍的方法不能在数学公式排版状态下使用。(如果要在数学公式排版状态下作类似的特征修饰,请参阅第六章)。

### 3.10 L<sup>A</sup>T<sub>E</sub>X 约定

L<sup>A</sup>T<sub>E</sub>X排版系统已为用户预先对在一定的文件版式下的全局定义好了许多参数。以下列举了其中一些基本约定:

- ① 版心正文部分的基准字号是10pt,如果用户设定为11pt,或12pt,那么基准字体只是相对基准字号的线度分别增大10%和20%。
- ② 在用户未对西文字体作任何说明时,系统将自动为你选取罗马字体。
- ③ 如果用户未在文件版式命令中对设置通栏还是双栏作任何说明,那么根据系统约定,为你选取是按通栏(单栏)纵向排版方式。

- ④ 页面文章所占版面(版心尺寸, 包括脚注, 但不包括页码) 为:

$$\text{高} \times \text{宽} = 345\text{pt} \times 528\text{pt}$$

(大约12.2cm×18.6cm)。

- ⑤ 除非用户自己专门设置了页码的起始数字和页码字体, 系统总是将页码从1 开始计数。对文章类(article) 或报告类(report) 文件, 页码被排在页面的下方居中; 而对于书籍类(book) 文件, 页码将根据单、双页的不同分别排在右上方和左上方(而另外一方则被放置章节的标题, 从而形成标准的书眉)。

\*

\*



W

24 A

\*

## 排版环境命令

本章介绍一些 $\text{\LaTeX}$ 系统专用的排版环境命令，主要包括引文、诗句、行对位、条目列表等排版环境。

### 4.1 独立排版环境的进入与退出

$\text{\LaTeX}$ 的排版环境是这样一个概念，许多特定的排版功能只有在特定的环境下才能实现。例如在许多文章中常常会出现一些引文，按照书刊印刷惯例，除了引文部分要取小一号字体或强调字体之外，其版面宽度也要比其它的正文部分略小一些，即版心的页边距要稍大些。要建立这样的独立排版环境，对 $\text{\LaTeX}$ 来说非常方便。与花括号分组控制命令设置排版小环境相类似， $\text{\LaTeX}$ 通常使用配对的开始( $\backslash\text{begin}\{$ 环境注解 $\}$ )与结束( $\backslash\text{end}\{$ 环境注解 $\}$ )环境的排版控制命令来实现这一目标。例如对引文，我们可在引文稿前后插入一对引文环境控制命令：

```
\begin{quotation}
  <引文部分>
\end{quotation}
```

这里 $\backslash\text{begin}\{$ 环境注解 $\}$ 和 $\backslash\text{end}\{$ 环境注解 $\}$ 实际上是确定了独立环境的上界和下界，在此环境中的各种排版信息和指令对外界不会产生任何影响。注意由于 $\text{\LaTeX}$ 对单个的回车符作空格处理，所以上面的引文环境控制命令中的第一行命令和最后一行以及引文的正文部分并不一定要分行输入。所以你也可将上面的命令用如下形式输入：

```
\begin{quotation} <引文部分> \end{quotation}
```

不过为了便源程序清晰易读，建议用户对环境控制命令采用分行输入方法。 $\text{\LaTeX}$ 系统提供了几十个特殊排版环境命令可供选择。比如列表环境、数学公式显示环境、引文环境和作图表环境等。每个环境的排版内容均由开始和结束两行控制命令括起来。在许多(但不是全部)特殊排版环境里可以再设置若干个次独立环境，并且还可逐级嵌

套。但是在使用时要对每一对环境控制命令的起始和结束位置都要安排妥当，切不可让几个独立环境的起始和结束命令位置之间发生交迭。例如下面的环境控制程序

```
\begin{环境注解1}
\begin{环境注解2}
\end{环境注解1}
\end{环境注解2}
```

在执行排版时就会出错。正确的排版程序应将最后两行对调一下。在插入顺次由大环境进入小环境的排版控制命令后，不要忘了先退出小环境然后再结束大环境的排版。由于 $\text{\LaTeX}$ 对所有文稿中额外插入的空格均会按省略规则处理，为了减少错误我们建议用户在设置独立环境控制命令时采用梯形嵌套法。下面是这种方法的示意图：

```
\begin{环境注解1}
  \begin{环境注解21}
    \begin{环境注解3}
      \end{环境注解3}
    \end{环境注解21}
  \begin{环境注解22}
    \end{环境注解22}
  \end{环境注解1}
```

最后我们还要再一次提醒读者，原来大环境中的特殊字体、字号等指令在进入小环境后立刻无效，所以外加的特殊排版控制命令一定要设置在小环境中。此外有些环境控制命令不能嵌套使用。例如文稿排版环境命令

```
\begin{document}
.....
\end{document}
```

就是一个不能嵌套使用的环境。本章的后面几节里将向读者着重介绍几种最常用的环境控制命令。

## 4.2 引文环境命令

引文(`quote`, 或`quotation`)环境控制命令根据具体的排版样式主要有两种形式，其环境注解词分别为“`quote`”和“`quotation`”。

### 4.2.1 命令格式1

引文环境控制命令的第一种格式为

```

\begin{quote}
  <引文部分>
  (空出一行)
\end{quote}

```

注意:

① 小号字的排版命令应加在引文环境里面。如果这个命令放在外面则对引文排版毫无用处。

② 引文部分的版面宽度比正常的版心宽度要小2厘米左右,即引文部分两边各向中间缩进1厘米。

③ 输出时引文中的每一段开头都是顶格排版;各段之间的区分是靠空出的半行来区分的;引文整体的开始行和结束行与上下文各空出一行。

④ 养成在引文最后空出一行的好习惯。尤其是在引文中变换了字体或字号后,这个空行对于完美的引文排版就显得重要。

⑤ 如果引文后面你想另起一段,不要忘记在\end{quote}下面再空出一行。否则后面一段起始的地方不会缩进排版。

⑥ 本环境命令适用于编排独立显示的较短引文。

### 4.2.2 命令格式2

引文环境控制命令的另外一种格式为

```

\begin{quotation}
  <引文部分>
  (空出一行)
\end{quotation}

```

这个环境命令的排版效果与第一种形式相比,其差别是在输出时引文中的每一段开头都不是顶格排的,即会缩进一定长度再排(缩排);此外段落之间不再另外空行。此环境命令适用于编排独立显示的较长的引文。

下面是一个先后使用上面两种格式的引文环境控制命令的排版源程序实例:

```

\documentstyle{article}
\begin{document}
Here are two paragraphs about Maine's woodland terrors,
from {\em American Myths and legends},
by {\sf Charles M. Skinner}.
\begin{quote}
\ldots It is feared that some of the creatures which infest
the woods of Aroostook, Piscataquis, and Penobscot counties,
especially in the lumbering season, have had their
mischievous qualities magnified in local myths for the

```

silencing of fretful children\ldots.

Like the duppies and rolling caves of the West Indies, these creatures are not seen as often as they were, for the have a fixed hostility to schools, never venturing within ten miles of one.

\end{quote}

Following these paragraphs, you will read another two paragraphs from the same book.

\begin{quotation}

The will-am-alone is a quick little animal, like a squirrel, that rolls in its fingers poison-lichens into balls and drops them into the ears and on the eyelids of sleeping men in camp, causing them to have strange dreams and headaches and to see unusual objects in the snow\ldots.

More odd than this animal is the side-hill winder, a rabbit-like creature so called because he winds about steep hills in only one direction; and in order that his back may be kept level, the down-hill legs are longer than the up-hill pair. He is seldom caught; but the way to kill him is to head him off with dogs when he is corkscewing up a mountain. As the winder turns, his long legs come on the up-hill side and tip him over, and easy prey.

\end{quotation}

\end{document}

输出结果:

Here are two paragraphs about Maine's woodland terrors, from *American Myths and legends*, by Charles M. Skinner.

... It is feared that some of the creatures which infest the woods of Aroostook, Piscataquis, and Penobscot counties, especially in the lumbering season, have had their mischievous qualities magnified in local myths for the silencing of fretful children....

Like the duppies and rolling caves of the West Indies, these creatures are not seen as often as they were, for the have a fixed hostility to schools, never venturing within ten miles of one.

Following these paragraphs, you will read another two paragraphs from the same



book.

The will-am-alone is a quick little animal, like a squirrel, that rolls in its fingers poison-lichens into balls and drops them into the ears and on the eyelids of sleeping men in camp, causing them to have strange dreams and headaches and to see unusual objects in the snow...

More odd than this animal is the side-hill winder, a rabbit-like creature so called because he winds about steep hills in only one direction; and in order that his back may be kept level, the down-hill legs are longer than the up-hill pair. He is seldom caught; but the way to kill him is to head him off with dogs when he is corkscrewing up a mountain. As the winder turns, his long legs come on the up-hill side and tip him over, and easy prey.

注意比较一下由两种不同的引文环境命令控制输出的排版结果。

### 4.3 诗句环境命令

诗句(verse) 环境命令是用来对西文中的诗歌及韵文进行排版, 当然也可用来编排其它特殊类型的分行式文体。这种环境命令可使 $\text{\LaTeX}$  在为你排诗句时, 每行均缩进一定的空白字距。但是分行的工作得由你自己用双反斜杠控制了。诗句环境控制命令的格式为

```
\begin{verse}
<第一行诗句> \\
<第二行诗句> \\
.....
<最后行>
\end{verse}
```

注意:

- ① 最后一行诗句不必要再加双反斜杠换行控制符。
- ② 某一行诗句太长时会自动换行, 并会相对将诗句拉齐的左边界再缩进一定距离打印。
- ③ 如果在诗文后面你想另起一段, 同样不要忘了在`\end{verse}` 下面再空出一行。否则后面一段起始的地方不会缩进排版。下面是关于诗句环境控制命令使用的一个完整的可执行程序实例:

```
\documentstyle[11pt]{report}
\begin{document}
There was a popular American song, named as {\em Soup Song},
in 1937, the words of which are
```

```

\begin{verse}
  I'm spending my nights at the flophouse,\\
  I'm spending my days on the street,\\
  I'm looking for work and I find none;\\
  I wish I had something to eat.\\
  Soup, soup, they give me a bowl of soup.\\
  Soup, soup, they give me a bowl of soup\ldots.
\end{verse}

The basic melody, however, is the copy of the Scotland song
{\em My Boonie}.
\end{document}

```

输出结果:

There was a popular American song, named as *Soup Song*, in 1937, the words of which are

I'm spending my nights at the flophouse,  
 I'm spending my days on the street,  
 I'm looking for work and I find none;  
 I wish I had something to eat.  
 Soup, soup, they give me a bowl of soup.  
 Soup, soup, they give me a bowl of soup....

The basic melody, however, is the copy of the Scotland song *My Boonie*.

## 4.4 行居中环境命令

行居中环境控制命令是将分离的长短不同的各行内容均根据版心的中心位置居中左右对称排版。这种控制命令有多行居中和单行居中两种形式。

### 4.4.1 多行居中命令

在行文中若要对多行文字采取居中(center)排版(例如要排连续几行居中的大小标题),则可按照如下命令格式设置居中环境:

```

\begin{center}
  <西文居中行1 内容> \\
  <西文居中行2 内容> \\
  .....
  <最后行>

```

```
\end{center}
```

注意:

- ① 欲居中的每一行末尾不要忘了加上换行控制符“\\”, 但最后一行不必再加。
- ② 当某一行在版心规定宽度内排不下用户要求居中的西文内容时, 系统会自动将该行内容拆成几行, 并仍保持分行居中的排版状态。

#### 4.4.2 单行居中命令

当用户想要居中的内容只有一行时, 除了采用第一种方法之外, 还可使用更简单的控制命令, 格式为

```
\centerline{西文居中行内容}
```

下面的排版程序实例展示了如何使用上面两种方法来设置行居中环境控制命令。

```
\documentstyle[11pt]{article}
\begin{document}
\begin{center}
{\large \bf Damping Effects on the Reflection of the Metallic\\
Superlattice for Soft X-rays and Extreme\\ Ultraviolet\\}
YAO He-sheng, FENG Wei-guo and WU Xiang\\
{\footnotesize \sf Pohl Institute of Solid State Physics,
Tongji University, Shanghai 200092, P. R. China}
\end{center}
Based on the hydrodynamic model theory and the transfer-matrix
method, we have reexamined the reflection properties by taking
account of damping effects to the system of the metallic
superlattice for the s-polarized soft x-rays and extreme
ultraviolet. For the normal incidence, we find that the
reflection intensity spectrum will decline monotonously
with the increasing of the retardation, but for the regions of
the higher frequencies and the smaller grazing angles, the
damping effects could be neglected. This work was published
on {\em Commun. Theor. Phys.} ({\bf 13}, 1990, 425--432).
This project was supported in part by \\
\centerline{\sf the Chinese National
Advanced Technology Foundation.} and part by\\
\centerline{\sf Department of Physics, Tongji University}
\end{document}
```

输出结果:

# Damping Effects on the Reflection of the Metallic Superlattice for Soft X-rays and Extreme Ultraviolet

YAO He-sheng, FENG Wei-guo and WU Xiang

Poh Institute of Solid State Physics, Tongji University, Shanghai 200092, P. R. China

Based on the hydrodynamic model theory and the transfer-matrix method, we have reexamined the reflection properties by taking account of damping effects to the system of the metallic superlattice for the s-polarized soft x-rays and extreme ultraviolet. For the normal incidence, we find that the reflection intensity spectrum will decline monotonously with the increasing of the retardation, but for the regions of the higher frequencies and the smaller grazing angles, the damping effects could be neglected. This work was published on *Commun. Theor. Phys.* (13, 1990, 425-432). This project was supported in part by

the Chinese National Advanced Technology Foundation.

and part by

Department of Physics, Tongji University

## 4.5 行右齐和行左齐环境命令

在一般情况下,  $\text{\LaTeX}$  会自动将你的正文部分中的各个段落的内容以版心的左右边界拉齐排版。但有些时候, 一些文件中的某个特殊的版面(例如西文信笺上顶部的地址部分) 只需要将各行的右边对齐(`flushright`) 或左边对齐(`flushleft`) (国内印刷业亦称“齐肩”), 对于这种情形, 可使用行右齐或行左齐环境控制命令。

### 4.5.1 行右齐命令

命令格式:

```
\begin{flushright}
  <右齐行1 内容>
  <右齐行2 内容>
  .....
  <最后行>
\end{flushright}
```

注意:

① 凡是在行尾加了换行控制符的各行内容均能向右边界靠齐。但是当一行内容太多排不下时系统会自动作拆行靠右齐处理, 此时左边会出现参差不齐样式。

② 在右齐环境中分段控制符号(空出一行)无效。要编排自然段则须先退出右齐环境,并空出一行,然后假如需要的话可重新进入行右齐环境。

③ 当用户想要按右齐方式排版的内容只有一行时,可以采用另外一种简单的单行右齐控制命令,其标准格式为

```
\rightline{西文右齐行内容}
```

#### 4.5.2 行左齐命令

命令格式:

```
\begin{flushleft}
  <左齐行1 内容>
  <左齐行2 内容>
  .....
  <最后行>
\end{flushleft}
```

注意:

① 在编辑源文件中靠左对齐的输入方式对本命令毫无影响。凡是在行中或行尾加了换行控制符的分出的各行内容均能向左边界靠齐。

② 在通常不进入行左齐环境的排版状态下,单靠双反斜杠换行控制符也可排成靠左对齐的版式,但是与行左齐命令控制下的环境相比,有以下区别。首先,当一行内容太多排不下时系统会自动作拆行处理,在行左齐环境下各行内容靠左对齐,这时右边会出现参差不齐的样式。而在通常情况下系统又会自动将各行两边撑足。其次在左齐环境中分段控制符号(空出一行)无效,所以要分节则须先退出左齐环境,再空出一行,然后假如需要的话重新再进入行左齐环境。再则新的一节开始的地方也不象在一般情况下会缩进一点距离开始排版。

③ 当用户想要按左齐方式排版的内容只有一行时,也可以采用简单的单行左齐控制命令,其格式为

```
\leftline{西文左齐行内容}
```

作为行右齐和左齐环境控制命令的使用实例,我们在下面程序中给出了寄往美国物理学评论(Physical Review)用 $\text{\LaTeX}$ 排版语言编辑的一封信的源文件的一部分。其中多次使用了行右齐和左齐环境控制命令。读者可对比程序后面附上的输出拷贝,从而可得到一些启发。

```
\documentstyle [11pt]{article}
\begin{document}
\begin{flushright}
Department of Physics\\Tongji University\\
```

```
Shanghai 200092\\P.\\,R.\\,China\\
Tel.: 5455080-3380\\ \\today
\\end{flushright}
```

```
\\begin{flushleft}
Dr. Peter D. Adams \\Editor \\
Physical Review B \\Editorial Offices \\
1 Research Road, P.\\,B.\\,Box 1000 \\
Ridge, New York 11961 \\U.\\,S.\\,A.
\\end{flushleft}
```

```
\\begin{flushleft}Dear Editor,\\end{flushleft}
```

```
\\begin{flushleft}
Enclosed please find three copies of our paper entitled
{\\bf Effect of plasma waves on the optical properties of
multilayered metallic Fibonacci quasi-superlattice} (PACS No.:
71.45.Gm and 68.65.+g) which was finished recently by professor
X. Wu, Mr. N.\\,H. Liu and myself. The prepared manuscript was
printed in the \\LaTeX{} style at ICTP, which is now available
in my disc. We will feel great pleasure if the article could
be published on the {\\em Physical Review B: Condensed Matter}
as a regular paper. If you have any question concerned with
the paper, please be kindly let me know.
\\Thank you!
\\end{flushleft}
```

```
\\end{document}
```

输出结果:

Department of Physics  
Tongji University  
Shanghai 200092  
P. R. China  
Tel.: 5455080-3380  
August 11, 1993

Dr. Peter D. Adams  
Editor  
Physical Review B

Editorial Offices  
1 Research Road, P. B. Box 1000  
Ridge, New York 11961  
U. S. A.

Dear Editor,

Enclosed please find three copies of our paper entitled **Effect of plasma waves on the optical properties of multilayered metallic Fibonacci quasi-superlattice** (PACS No.: 71.45.Gm and 68.65.+g) which was finished recently by professor X. Wu, Mr. N. H. Liu and myself. The prepared manuscript was printed in the  $\text{\LaTeX}$  style at ICTP, which is now available in my disc. We will feel great pleasure if the article could be published on the *Physical Review B: Condensed Matter* as a regular paper. If you have any question concerned with the paper, please be kindly let me know.

Thank you!

## 4.6 引文环境中的行对齐命令

引文环境控制命令实质上是设置了多行的排版盒子，其特征是版面的左右两边距版心边界额外空出了一些白边。如果在此环境中使用行右齐和行左齐控制命令：

`\leftline{...}`            `\rightline{...}`

那么会出现部分文字超越引文边界，与版心边界拉齐的排版结果。如果需要在引文环境中设置与引文边界对齐或居中的版面，除了嵌套使用行对齐排版环境之外，还可以使用另外几条行对齐命令：

- `\centering` 在引文环境或其它类似的小页排版盒子(参见第5.8.1节)中插入此命令，则可控制以后的分行文字(用“\\”分隔开)居中排版。这个命令常被用在插图和表格环境中。
- `\raggedright` 控制此命令后面的分行文字以引文或小页排版盒子的左边界对齐排版。排版效果类似`flushleft`环境。
- `\raggedleft` 控制此命令后面的文字以引文或小页排版盒子的右边界对齐排版。排版效果类似`flushright`环境。

以上三个命令在退出当前排版环境后立即无效。

## 4.7 抄录环境命令

假如你是在计算机上用全屏幕文本编辑软件输入你的排版源程序，那么你在终端屏幕上看到的由键盘逐字逐句打入的文件字体是属打字机体类型的。即每个字有固定

的宽度, 源文件显示的样式与排版后的形式完全两样。此外, 源文件中总会夹杂着不少控制指令, 输出时, 这些指令不会打印出来。但有时我们又偏偏就是要这种样式的源文件用打字机字体抄录(verbatim)到输出拷贝上。例如我们需要在一篇文章中的某个篇幅内原封不动地打印出一个 $\text{\LaTeX}$ 源文件的清单, 并且希望其中的一些指令在排版时失效, 只是作为一般打印机字体的行文处理(如本书中罗列的许多程序)。 $\text{\LaTeX}$ 语言中的抄录控制命令就是为进入和退出这样一种特殊排版环境而设计的, 其作用实际上就是将 $\text{\LaTeX}$ 关闭一段时间, 将用户输入的处在抄录环境中的小样文件不折不扣地用打字机字体打印出来。抄录控制命令亦有两种形式: 多行和单行抄录环境命令。

#### 4.7.1 多行抄录环境命令

第一种命令格式主要用于设置较大信息量的多行源文件的逐行逐句拷贝环境。命令格式如下:

```
\begin{verbatim}
<无须重新排版的多行正文>
\end{verbatim}
```

注意:

① 所有 $\text{\LaTeX}$ 语言中的控制命令在此环境中毫无作用, 即此命令可用打印机字体打印出所有的排版命令, 但是环境结束控制命令“`\end{verbatim}`”除外。

② 输出时在此环境中的所有内容将按源文件原样拷贝, 包括所有空格、空行、换行形式和控制字符等。

例如程序

```
\begin{verbatim}
\small \bf This is \today.
\end{verbatim}
```

会输出

```
\small \bf This is \today.
```

#### 4.7.2 单行抄录命令

第二种命令格式主要用于设置仅限于一行内的小信息量源程序拷贝排版盒子。其命令格式如下:

```
\verb<边界标识字符><无须重新排版的字符串><与前面相同的边界标识字符>
```

注意:

① 在控制指令“`\verb`”后面出现的任意第一个字符起到了关闭 $\text{\LaTeX}$ 的作用(类似于第一种格式中的“`\begin{verbatim}`”), 同时也为后续的无须排版的字符串设定上边



界。此后当前面第一个字符再次出现时, 逐字拷贝的抄录环境即告结束。换句话说, 前后出现的两个相同的任选字符实际上充当了带参量的花括号角色。与花括号的区别仅在于配对花括号是左右两个完全对称的字符。为了不至于与其它边界控制符混淆, 建议使用“|”、“\*”和“+”等不分左右的字符来充当边界标识字符。

② 在上面两个相同的任意字符中间的任何其它控制字符(包括反斜杠换码符) 和控制词类命令均可直接打印, 唯独不能打印任意选定的边界字符。

例如我们可取数学中的绝对值符号“|”(键盘上的竖线字符) 充当取定的第一个(也是命令格式中的最后一个) 字符, 那么假定想要打印日期命令“\today”, 则可输入

```
\verb|\today|
```

当然, 如果你选取字符“+”或“\*”来取代“|”也有同样功效。例如需要打印“|x|”可使用命令

```
\verb+|x|+
```

读者应注意空格也是字符。当 $\text{\LaTeX}$ 执行排版过程中, 上述两种格式的抄录命令在处理源程序里的空格时会特地在版面上留出符合打字机字体规范的一个字宽。但在有些计算机程序设计语言中, 希望将源程序中必须加进的所有空格用示意符号“\_”来表示。遇到这种情形, 可使用带星号的抄录环境命令:

```
\begin{verbatim*}
<无须重新排版的多行正文>
\end{verbatim*}
```

或单行抄录命令

```
\verb* <边界标识字符><无须重新排版的字符串><与前面相同的边界标识字符>
```

## 4.8 条目列表环境命令

所谓条目, 是指出现在有些特殊版面上排列比较规则的可编号的或无须编号的块状行文。例如“协议”、“条约”、“产品技术参数说明书”和“合同书”等类型的文件就会出现含有若干条目的版面。条目列表时, 习惯上总希望各个条目之间略微空出点纵向间距; 左、右的页边距与正文要有所差别; 主条目与子条目的列表编号法应有明显区别。然而, 对于使用 $\text{\LaTeX}$ 排版系统的用户来说, 完全不需要去考虑这些细节。因为 $\text{\LaTeX}$ 已为用户提供了足够使用的常规的三种条目列表环境: 编号条目列表环境, 无序号条目列表环境和描述型编码条目列表环境。

### 4.8.1 编号条目列表环境

编号条目列表环境命令的一般格式为:

```

\begin{enumerate}
\item_1 <条目1>
\item_2 <条目2>
.....
\item_n <条目n>
\end{enumerate}

```

说明:

- ① 各条目正文部分会自动成矩形块状缩进齐肩排版。
- ② 每一个条目必须要由条目控制命令“\item”引出。在输入源文件时，与条目正文之间至少要空出一个空格。每一个条目正文输入完毕时无须作任何强迫换行记号(如双反斜杠换行控制符“\\”或“\par”等)。
- ③ 此命令可以自身继续嵌套使用，但是系统最多只能编排四个级别的条目。在不同级次的列表环境中，系统会自动切换各子条目的数字编号形式(例如一级条目采用的编号为阿拉伯数字1, 2, 3, ...; 二级条目为英文小写字母a, b, c, ...; 而第三级次的则是罗马小写数字i, ii, iii, ...; 等等)。

例如源程序:

```

\begin{enumerate}
\item Introduction
  \begin{enumerate}
\item A laser for the x-ray region
  \begin{enumerate}
\item Background
\item The data base
\item The laser medium
\end{enumerate}
\item Definitions and optical properties unique
      to the region of interest
  \begin{enumerate}
\item Wavelength ranges defined
\item Filters and reflectors
\end{enumerate}
\end{enumerate}
\item Principles of Short Wavelength Lasers
\item Pumping by Exciting Plasma Ions
\end{enumerate}

```

可以输出:

1. Introduction
  - (a) A laser for the x-ray region
    - i. Background

- ii. The data base
  - iii. The laser medium
- (b) Definitions and optical properties unique to the region of interest
  - i. Wavelength ranges defined
  - ii. Filters and reflectors
- 2. Principles of Short Wavelength Lasers
- 3. Pumping by Exciting Plasma Ions

#### 4.8.2 无序号条目列表环境

无序号条目列表环境命令的一般格式为:

```
\begin{itemize}
\item_1 (条目1)
\item_2 (条目2)
.....
\item_n (条目n)
\end{itemize}
```

说明:

① 无序号条目列表环境中各个条目中正文部分排版的样式与编号条目列表环境命令中的完全一样。只是在输出时,各条目不再编号,而是排在相同的特殊标记后而引出(如实心圆点“•”,起迄符“-”等)。

② 此命令可以自身继续嵌套使用。系统会自动切换各子条目前的特殊标记以区别各子条目的级别。

③ 此命令亦可以与上面介绍的编号条目列表环境命令相互嵌套使用。排版时,各个子条目前的编码形式按各自所处的排版环境而定。

例如源程序:

```
\begin{enumerate}
\item Introduction
\begin{itemize}
\item Finite sums
\begin{itemize}
\item Sums of power of natural numbers
\item Sums of reciprocals of natural numbers
\item Sums of the binomial coefficients
\end{itemize}
\item Numerical series and infinite products
\begin{itemize}
\item Examples of numerical series



```

```

\item Examples of infinite products
\end{itemize}
\end{itemize}
\item Functional Series
\end{enumerate}

```

可以得到硬拷贝:

#### 1. Introduction

- Finite sums
  - Sums of power of natural numbers
  - Sums of reciprocals of natural numbers
  - Sums of the binomial coefficients
- Numerical series and infinite products
  - Examples of numerical series
  - Examples of infinite products

#### 2. Functional Series

### 4.8.3 描述型编码条目列表环境

有些条目并不是由序号或特殊字符标记引出的。例如在任何一本西文字典中, 每个条目是以单词(通常用黑体印出)为首来编排的。又如在一份仪器说明书中要对一些重要的器件名字作详细解释, 那么在排版时就得将这些器件名字编排在每一项说明文字的最前面。对这一类条目, 我们称之为描述型编码条目。设置描述型编码条目列表环境, 除了要令系统对每个条目的正文部分安排适当的版面之外, 还要适当地留出一些条目前面的编码位置, 从而提供给用户为条目“贴”标签(描述型编码形式或条目名称)。描述型编码条目列表环境的控制命令为:

```

\begin{description}
\item[ 标签1] <条目1>
\item[ 标签2] <条目2>
.....
\item[ 标签n] <条目n>
\end{description}

```

说明:

① 每个条目前的标签参量可以是任意行文状态下的合法字符串(注意不要忘记用方括号夹起来)。

② 由于这里采用方括号来设置标签参量环境, 所以不能再简单地使用方括号作为字符串来作为你的描述型编码。如果你确实需要用方括号夹带阿拉伯数字来作为条

目序号(例如参考文献的序号),那么可以另外用一对分组控制符“{}”将方括号与数字括起来后,再作标签使用。

③ 此命令同样可以自身多次嵌套使用,也可与上面介绍的编号条目列表环境命令和无序号条目列表环境命令相互之间嵌套使用。排版时,各个子条目前的编码形式按各自所处的排版环境而定。

例如有一个小小的通讯录源程序:

```
\begin{itemize}
\item FENG Weiguo
  \begin{description}
    \item [Address:] Department of Physics, Tongji
      University, 1239 Siping Road, Shanghai 200092,
      P. R. China
    \item [Tel.:] 5455080-3796
    \item [Telex:] 33488 TJDC CN
  \end{description}
\item MO Huilin
  \begin{description}
    \item [Address:] Dean's Office, Tongji University,
      1239 Siping Road, Shanghai 200092, P. R. China
    \item [Tel.:] 5455080-2872
    \item [Telex:] 33488 TJDC CN
  \end{description}
\end{itemize}
```

可以得到硬拷贝:

- FENG Weiguo

Address: Department of Physics, Tongji University, 1239 Siping Road, Shanghai 200092, P. R. China

Tel.: 5455080-3796

Telex: 33488 TJDC CN

- MO Huilin

Address: Dean's Office, Tongji University, 1239 Siping Road, Shanghai 200092, P. R. China

Tel.: 5455080-2872

Telex: 33488 TJDC CN



## 5

# 版式处理命令

本章详细介绍如何调用 $\text{\LaTeX}$ 版式处理命令来实现在一篇比较规范完整的西文文章中的章节、脚注、目录、和附录等版式结构的自动排版。

## 5.1 章节排版命令

如何将你的文件编排章节(包括小标题、字体和字号等),并将其按不同层次自动编号,首先取决于你在文件格式中所选的类型参量。例如对于报告类(report)文件, $\text{\LaTeX}$ 可为你处理如下几个层次的版面:章(chapter)、节(section)、款(subsection)、项(subsubsection)、段(paragraph)、小段(subparagraph)。文章类(article)文件除了缺少章次(chapter)结构外,其它结构与报告类文件完全相同。书籍类(book)文件则在报告类文件的章节结构基础上再增加“篇”(part)结构。

在排版中要实现这些结构是靠转码符“\”与上面提到的合法的英语章节款项等单词组成的控制命令再加上标题参量(不带序号的西文标题)来控制的。表5.1列出了书、报告和文章三大类文件中的所有可使用的未带参量的 $\text{\LaTeX}$ 章节排版控制命令(表中最后一栏列出的字号大小是根据基准字号为10pt的情形给出的)。章节排版命令的使用很简单,只要在每一个章节层次开始的地方加入上表中的控制命令,并且将该层次的西文标题行中的内容(不用考虑章节序号)用花括号括起来作为此命令的参量。这样系统就能根据章节层次的标准结构自动将参量选取合适的字体、相应的字号和格式安排在版面上,同时会替你将从前面延续到此命令处正确的章节序号补在标题的前面。后续的文稿将作为该层次的内容进行排版,直到出现新的章节排版命令为止。例如在报告类(report)文件格式下,某一章的英文标题为“Introduction for  $\text{\LaTeX}$  System”(  $\text{\LaTeX}$  系统介绍),你可在此章开始的地方输入

```
\chapter{Introduction for \LaTeX}
```

假如文件执行到此处时系统计数器算得这一章序号为7,那么你在输出时能看到如下排版样式:

表 5.1 章节排版命令表

章节层次	书(book)	报告(report)	文章(article)	标题字号(pt)
篇	<code>\part</code>	—	—	25
章	<code>\chapter</code>	<code>\chapter</code>	—	20
节	<code>\section</code>	<code>\section</code>	<code>\section</code>	17
款	<code>\subsection</code>	<code>\subsection</code>	<code>\subsection</code>	14
项	<code>\subsubsection</code>	<code>\subsubsection</code>	<code>\subsubsection</code>	12
段	<code>\paragraph</code>	<code>\paragraph</code>	<code>\paragraph</code>	10
小段	<code>\subparagraph</code>	<code>\subparagraph</code>	<code>\subparagraph</code>	10

## Chapter 7

# Introduction for $\text{\LaTeX}$

又如某一节的英文小标题为“The Quote Environment”(引文环境), 同样地你可在本节开始的地方输入

```
\section{The Quote Environment}
```

如果它是第七章的第三节的话, 你将获得如下拷贝,

### 7.3 The Quote Environment

$\text{\LaTeX}$  在执行你设置的章节排版命令时另外还在系统里自动记下了(记录在硬盘的某个文件上) 各个章节标题的全部内容, 并录下了排版至该命令时所处的页码, 这样为以后编目录作好了数据准备, 你就无须再为这些编辑中的琐碎小事伤脑筋了。

在后面, 我们将详细介绍如何编排相应的文件目录。有时作者希望在目录中出现的章节标题与正文中实际排版后出现的标题内容不完全一样, 对于这种情况,  $\text{\LaTeX}$  亦为你设计了一种带二参量的控制命令形式, 其一般格式为,

```
<章节排版命令> [目录中出现的西文标题] {正文中出现的标题}
```

若以上面第二个例子为例, 假如我们希望在目录中只出现“QUOTE”, 而在正文中维持上述标题样式, 那么我们将控制命令改成

```
\section[QUOTE]{The Quote Environment}
```

如果你不喜欢系统在某一层次的章节排版命令中为你选择的字体、字号以及计数



的形式, 而是想换一种格式, 那么可在这一层次的章节命令中(在左花括号之前) 插入一个省略编号标识符“\*”, 并在花括号内注上你所要的样式。仍以前面第二个例子为例, 假如我们希望将标号改成英文小写字母(a, b, c, d, ...), 且标题居中排版并采用基准字号(normalsize) 的等线体字体(sans serif), 那么我们可输入小样:

```
\section*{\centerline{\normalsize \sf
c. The Quote Environment}}
```

那样的话, 输出时你将发现如下结果:

c. The Quote Environment

说明:

① 用章节控制命令排版的章节标题不会出现“背题”。所谓背题是指在排版时, 标题恰好处在页面的底部(最后一行)。在标题下面已无法排文字, 或勉强挤进一或两行文字。换句话说, 标题已背离了文字。遇到这种情况, 根据规范的排版要求, 应将标题另起面排版。L<sup>A</sup>T<sub>E</sub>X 即具有这种功能。只要用户是按照本节介绍的排版控制命令去编排章节标题, 就不会出现令人感到极不舒服的背题情形。

② 各级章节标题所采用的字体均为黑体, 而所选字号则根据标题的级次逐级缩小(见表5.1)。

③ 使用章标题命令“\chapter”时, 对于双面印刷的版式twoside 情形, 系统会将每一章均由奇数页起排。如果某一章结束时不是落在偶数页, 那么系统会自动空出一页(即所谓背面空白) 以保证单数页起排。

④ 如果某一章中引用了相当数量的图表, 那么, 建议在每一章结束位置选用下述换页命令:

```
\clearpage           \cleardoublepage
```

以保证图表在该章范围内全部打印完毕; 即使在正文中不能排完的, 也要留出专门的页面完成图表的排版。后面一条命令“\cleardoublepage” 与前者的差别在于: 在双面印刷的版式中, 换页以后要保证由奇数页起排; 如果换页后落在偶数页, 那么系统会再空出一页。

## 5.2 脚注命令

写文章和编书总免不了标记脚注。所谓脚注, 是指正文中某个部分需要另外添加的注释。这个注释通常就安排在当前页的左下方, 而文中引用的地方只是标记上与脚注行首处相同的序号。L<sup>A</sup>T<sub>E</sub>X 为用户提供了非常方便的脚注排版控制命令, 其形式如下:

```
\footnote[脚注序号]{脚注内容}
```

说明:

① 系统会自动选择脚注所用的字号(8pt, 相当中文的六号字) 和字体(一般为罗马字体)。当你的文件中需要加脚注时, 你只须在文中引用脚注的位置插入包括完整脚注内容(以参量形式输入) 的控制命令。那么在 $\text{\LaTeX}$  为你执行排版后, 你就可在引用脚注页的左下方看到一条直线, 在它下面, 便是你需要的脚注。例如脚注程序<sup>1</sup>

```
\footnote{This is footnote example No.1.}
```

就可以得到本页底部的第一个脚注。当该页有若干个脚注时, 几个脚注会依次在直线下方排齐编号出现。

② 参量脚注序码 是提供给用户为某个脚注设置特定的编码用的选择参量, 所以你可以用此参数自行为脚注编号。此选择项可以省略。省略时系统用阿拉伯数字为你给脚注编号。例如程序<sup>20</sup>

```
\footnote[20]{This is footnote example No.2.}
```

则会输出如本页底部号码为“20” 的脚注。

③ 脚注命令可以在小页排版环境中使用(见本章第5.8.1 节), 但不能在小页排版盒子(参见第5.8.2 节) 或LR 排版盒子命令中使用。如果你一定要在这些排版环境中使用脚注, 那么还可以输入另外一对必须联合使用的等价脚注命令:

```
\footnotemark[脚注序码]
\footnotetext[脚注序码]{脚注内容}
```

第一行是脚注序码命令, 仅在需要引用脚注的地方加上一个脚注序码, 但不给出任何脚注正文信息。第二行命令为脚注正文内容的排版命令, 它给出页面底部脚注的具体内容, 但是在版心正文中引用该脚注的地方未作任何序码标志。例如程序

```
\fbox{This is a {\tt footnotemark}
\footnotemark[30]
example.}
\footnotetext[30]{This is footnote example No.3.}
example.}
```

会在正文版面上输出如下拷贝:

This is a footnotemark<sup>30</sup> example.

在执行程序中第二行脚注序码命令时会在“footnotemark” 单词的右上方加上脚注序码标志“<sup>30</sup>”, 而执行第四行脚注正文内容的排版命令时则控制输出如该命令出理时所在页面底部号码为“30” 的脚注。注意在程序中我们将命令“\footnotetext” 写在了LR 盒子命令“\fbox” 的外面。如果写在内部, 那不会得到脚注的任何内容。

<sup>1</sup>This is footnote example No.1.

<sup>20</sup>This is footnote example No.2.

<sup>30</sup>This is footnote example No.3.

④ 以上脚注的所有序码形式均为阿拉伯数字，如果需要改变序码的计数字体形式，可参阅10.5.4 小节。

### 5.3 旁注命令

旁注(marginal note)是指编排在版心之外靠近切口的注释文字。通常用于指出该页正文中直接有关的参考文献，或试图加以解说、详述、评论，以及进行引用。国外出版商常使用这种方式编排教科书和商品的说明书。与脚注同样，旁注仅限于在某一页需要直接引起读者注意时应用。旁注命令的标准格式为：

```
\marginpar[左页旁注]{右页旁注}
```

说明：

① 由于旁注规定排在书刊的切口一侧，所以旁注命令对单面或双面打印的排版方式不一样。对于单面排版打印方式，不论是单页或是双页，旁注一律排在右侧(左页旁注可省略)。对于双面排版印刷方式，每逢单页情形(亦称右页)，系统选择命令中的右页旁注，并将其排在版心右侧。对于双页情形(亦称左页)，系统选择命令中的左页旁注，并将其排在版心左侧；如果用户省略左页旁注，则系统改选右页旁注，并将其排在版心左侧(亦是切口处)。

② 旁注编排的位置(版面纵向行位置)将取决于命令所处的位置，一般旁注被设置在命令所在行的版心某一侧起排。

③ 由于旁注占据了版面的一定宽度，所以相应的版心宽度要适当缩小。这样的排版样式会浪费一些版面，使得一本书或报告所需的页数比无旁注的情形多得多，因而成本价格会上升。如果不是重要的说明书或是解说书，一般不要采用这种排版方式。

以下是旁注的一个排版实例(假定版心宽度只有8厘米)：

```
If you have a copy of the file \mbox{\tt sample.tex} in your
current directory; you can edit it just like any other ASCII
file. Then, run \LaTeX\ on the file \mbox{\tt sample.tex} by
typing: \marginpar{\em How to run the sample file }
\begin{verbatim}
        latex sample
\end{verbatim}
When \LaTeX\ has finished, it will have produced the file
\mbox{\tt sample.dvi} in your directory.
\marginpar [\em Here is the leftside note]
        {\em How to print the sample file}
You can then print this file on the Epson LQ Printer
easily by typing the command ``\verb|dvielq sample|''.
```

执行排版后的输出硬拷贝为:

If you have a copy of the file `sample.tex` in your current directory; you can edit it just like any other ASCII file. Then, run  $\text{\LaTeX}$  on the file `sample.tex` by typing:

```
latex sample
```

*How to run the  
sample file*

When  $\text{\LaTeX}$  has finished, it will have produced the file `sample.dvi` in your directory. You can print this file on the Epson LQ Printer easily by typing the command "`dvielq sample`".

*How to print the  
sample file*

## 5.4 页码和书眉排版命令

对于一般的不太长的西文文件, 用户不必在页面设计中的页码和书眉上多费精力。因为系统对于没有设置此类命令的源文件会自动将页码安排在版心的底部居中排版。但是假如遇到编排较长的报告和书之类的文件, 就不能不考虑页码和书眉(总称页眉)的版式了。所谓书眉, 是指为了便于检索, 在版面顶部印有书刊名称、篇章名或卷、期次、作者和篇名的内容。由于排版时, 书眉和页码常常排在一起, 所以一并称之为页眉(head)。

关于页码的版式安排主要牵涉到两个问题, 一个是选择什么样的页码数字字体, 另一个是在什么位置打印页码。前面一个问题好办, 后一个问题则要视具体情况而定, 因为它的位置必须得由书眉的排版方式来确定。所以 $\text{\LaTeX}$ 对于第一个问题设计了一个页码数字字体命令, 而在处理第二个问题时, 将页码打印位置和书眉版式的控制均由一个页眉版式命令来完成。

### 5.4.1 页码数字字体命令

正规西文书籍的印刷时, 为了便于区分, 扉页和目录部分页码(page number)数字的形式和书中正文部分页码的阿拉伯数字是不相同的。通常是以罗马数字形式出现。这样就有必要对页码数字字体的选择加上注解。 $\text{\LaTeX}$ 为用户提供了加下带单参量的页码数字字体控制命令:

```
\pagenumbering{ 页码参数 }
```

可供选择的页码参数有五种, 实际的什数形式和打印效果均被列在表5.2中。读者在使用页码数字字体命令时, 可从表5.2中第二栏里选取你所需要的英文参数。当你在源文件中的任意位置输入页码字体命令时, 当前页和以后的各页便会立即变换到以你新指定的页码数字形式分页, 并会无条件从一开始重新计数。例如当你在排版文件中输入了以下的页码版式控制命令:

表 5.2 页码形式参数表

页码形式	页码参数	实际计数和打印样式
阿拉伯数字	arabic	1, 2, 3, 4, 5, 6, 7, .....
罗马小写数字	roman	i, ii, iii, iv, v, vi, vii, .....
英文小写字母	alph	a, b, c, d, e, f, g, .....
罗马大写数字	Roman	I, II, III, IV, V, VI, VII, .....
英文大写字母	Alph	A, B, C, D, E, F, G, .....

`\pagenumbering{Roman}`

你就会从排版文件中页码版式控制命令生效的那一页起看到用罗马大写数字形式计数的页码。假如在你整个源文件中无任何页码命令，那么系统会自动按省略方式以阿拉伯数字形式编排页码。

#### 5.4.2 页眉版式命令

在打印机最终输出的每一个版面上不外乎三个主要部分：顶部(书眉部分)、中间(正文版心部分)和底部(脚注、页码部分)。源文件中的文件格式命令确定了这三个主体在版面上的空间大小。页码和书眉版式(page style)命令则是用来控制每一页的页码和书眉(假如需要的话)的排版方式。例如你是将页码印在底部还是在顶部，书眉的单双页形式如何安排等。为此目的 $\text{\LaTeX}$  替用户准备了一个带单参量的页面风格控制命令：

`\pagestyle{页眉排版方式}`

系统提供了四种可供选择的的标准页眉排版方式参量：

① **plain** 页码印在版心底部的中心位置。这种打印方式与文命类(article) 或报告类(report) 源文件中在缺省页眉版式命令时的页码打印方式相同。

② **empty** 本页既不打印页码，也不打印书眉，但系统仍在计页码数。

③ **headings** 页码打印在版心顶部，并根据源文件中的文件格式命令将章节标题作为眉注形式与页码共同排版，面底部则空白。例如对于“twoside”版式注解(或书籍类格式文件)，单数篇眉(节次标题)排在版心顶部右侧，双数篇眉(章次标题)排在左侧。

④ **myheadings** 若无其它书眉版式命令伴随其后时打印效果同参量“headings”效果相同。但是当后面紧跟“\markleft”、“\markright”和“\markboth”三个书眉版式命令中的任何一个时，则将按具体的书眉版式命令描述的方式去排版。下面对三个书眉版式命令分别作一些具体说明：

- `\markleft`

命令格式：`\markleft{左页书眉内容}`

左页书眉内容 是用户指定的出现在双数页(习惯上称为左页) 顶部的篇眉文字。省略时左页书眉为各章的标题文字。

- `\markright`

命令格式: `\markright{右页书眉内容}`

右页书眉内容 是用户指定的出现在单数页(习惯上称为右页) 顶部的篇眉文字。省略时右页书眉为各节的西文标题。

- `\markboth`

命令格式: `\markboth{左页书眉内容}{右页书眉内容}`

左页书眉内容和右页书眉内容 是用户指定的分别出现在双数页和单数页顶部的篇眉文字。

以上有关页眉的版式命令在源文件中一旦出现, 就对后续内容的页眉排版方式发挥作用。但是如果你只须变换某一页的页眉版式, 那么将页面风格控制命令改成如下形式:

`\thispagestyle{页眉排版方式}`

就可以了。命令中的页眉排版方式参量的选择如同以上所述的完全一样, 差别在于这里只对命令所在页的页眉排版方式起到控制作用。

## 5.5 附录版式命令

不同于本章第一节中的章节排版命令, 附录(appendix) 版式命令只是一个不带参量的简单控制命令。如果你准备在源文件某处开始排附录, 那么只要键入一个无参量的附录版式命令“`\appendix`”, 那么系统就会将后续的各个章节重新按附录版式来排版。对于书籍类(book) 或报告类(report) 的文件, 则在附录版式命令以后出现的每一章的标题上面一行将冠以单词“Appendix”(附录) 和大写英文字母序列(A, B, C, D, ……) 作为编号, 而各个层次的小标题则是以这些大写英文字母带头缀以小数点后, 再以常规的阿拉伯数字系列来逐层依次编号; 对于文章类(article) 文件, 则在附录版式命令以后出现的每一节标题的行首仅以大写英文字母序列(A, B, C, D, ……) 作为编号。假如你要系统为你编排目录, 那么在目录中将会以同样的附录标题形式出现。例如你想在一个报告(report) 类源文件末尾加一个西文字体表附录(List of Fonts), 那么下面程序可为你实现这一目的。

```
\appendix
\chapter{List of Fonts}
```

输出结果:

**Appendix A**  
**List of Fonts**

假定其中第三节的英文小标题为“The Roman type style”(罗马字体),那么你可按照下面程序中的排版方式来输入源文件。

```
\section{The Roman type style}
```

输出结果:

### A.3 The Roman type style

如果你想在附录中的某一部分更换一种你自己设计的标题编号格式,那么可在这部分中的所有的章节命令中(在左花括号之前)插入一个省略编号标识符“\*”,并在相应的花括号内注释你所要的样式。

## 5.6 列目录命令

有了前面几节的充分准备,打印目录(Contents)已不是一件难事了。因为假如你是按照上面几个标准的版式命令去编辑你的源文件的话,那么系统在对你的文件执行排版时就会自动记录下所有关于各章节的目录信息(页码和标题)。问题是你想不想要一个目录。如果你确定要排一个目录(比如你想编本书或写一份较长的报告),那么你可键入如下命令:

```
\newpage  
\tableofcontents
```

其中第一行命名“\newpage”(另面命令)是为了美观而令目录另起一页排版。 $\text{\LaTeX}$ 在首次读到排目录命令“\tableofcontents”时,就会在硬盘里建立一个与AUX文件类似的但扩展名为[.TOC]的有关目录格式的信息文件,同时在命令所在的位置编排目录。然而你得明白第一次获得的目录是空的,这是因为在你运行 $\text{\LaTeX}$ 之前,硬盘内还没有建立起任何版本的有关你文件目录格式的信息文件。但当你再次运行 $\text{\LaTeX}$ 时,系统就会根据以前执行排版时在AUX文件和TOC文件中记下的有关信息为你在目录页内进行逐行格式化排版。以后每逢要最后确定一个新版本文件,原则上应该运行 $\text{\LaTeX}$ 两遍。否则你在目录中看到的只是上一次编辑的老版本结果。

## 5.7 题名排版命令

在标准化术语中,称图书的书名、期刊的刊名、文章的篇名以及论文或报告题目等为题名(title)。题名的作用在于指明需要排版的文件主题。用来登录题名、作者的姓名(author)、撰写文件的日期(date)以及其它有关说明文字的独立印刷页称为题名页(title page)。在 $\text{\LaTeX}$ 排版系统中,你不必煞费苦心专门去设计这样一个特殊的题名版面或题名页,因为 $\text{\LaTeX}$ 已经替用户准备了编排题名用的两大类命令:题名信息存贮命令和题名排版命令。

第一类命令主要有三条, 是用来存贮题名有关信息。控制命令通常是放在源文件的正文前面的全局说明区域位置(位于文件格式命令和文稿排版环境起始命令两行之间), 其功能仅仅是为用户在内存里记录文件的题目(title)、作者的姓名(author)以及撰写文件的日期。如果你没有输入另外的题名排版命令, 在排版输出时是不会有文件题名的信息打印出来的。

记录文件题目、作者姓名和撰稿日期的三个带单参量控制命令的基本格式分别如下所示:

```
\documentstyle...
.....
\title{ 西文文件题目}
\author{ 作者西文姓名}
\date{ 西文日期}
\begin{document}
.....
```

注意:

① 如果你要让 $\text{\LaTeX}$ 为你排题目, 前面两个存贮信息用的控制命令不能缺少。即使你不想打印作者姓名, 你也应输入缺省参量命令“ $\backslash\text{author}\{\}$ ”。

② 如果你未输入日期命令“ $\backslash\text{date}\{\text{西文日期}\}$ ”,  $\text{\LaTeX}$ 会自动用命令“ $\backslash\text{today}$ ”来替代。

③ 上述三个控制命令中的参量的行文排版格式可任意选择。你可将文件题目用换行命令“ $\backslash\backslash$ ”分成若干行; 当作者名字较多时, 你也可用换行命令作分行处理, 甚至添上每个作者的单位; 此外西文日期的格式亦不必严格按照标准的形式写(例如你可以只写年月)。

④ 如果题名页需要添加脚注(例如编排致谢之类的信息), 用户还可使用再增添一条致谢命令:

```
\thanks{ 致谢类文字}
```

其中致谢类文字 可以是对赞助者致谢词、作者地址或其他脚注文字。

另外一类命令只有一条, 它是一种不带参量的题名排版命令, 其功能只是在插入命令的地方编排题名部分的信息。如果你在源文件最前面的全局说明区域已经输入了几个记录题名版面信息的控制命令, 那么你就可以选择合适的位置插入 $\text{\LaTeX}$ 题名排版控制命令

```
\maketitle
```

注意:

① 按照一般书刊印刷习惯, 题名通常排在目录之前。所以如果你欲将题名置于输出文件之首, 那么你可在文件起始命令“ $\backslash\text{begin}\{\text{document}\}$ ”后面一行立即输入命令“ $\backslash\text{maketitle}$ ”, 紧接着再输入有关的列目录命令。



② 在实际输出时各行内容均会居中排版打印, 打印的顺序是题目、作者署名、日期。对于文章类(article) 源文件, 题目的内容不会单独另页排版, 而是被排在正文第一页版心的最上面。对于书类(book) 和报告类(report) 文件则会专门留出一页作为题名页(扉页) 来排版。

③ 如果你不想让文章类(article) 的题名排在正文第一页的上方, 即你也想让题名另页排版, 那么你得在源文件最前面的文件格式命令中方括弧内加进一个题名页版式注解(titlepage) 例如对一个11 点(pt) 的文章类(article) 文件, 你可将文件格式命令写成以下形式:

```
\documentstyle[11pt,titlepage]{article}
```

或者对题名部分的内容设置一个题名页排版环境:

```
\begin{titlepage}
.....
\end{titlepage}
```

在这个排版环境中, 用户可根据需要自行设计题名页的版式。

④ 如果你不喜欢系统为你选择的字体或字号, 那么你可在第一类各命令中的参量花括号内加注一些字体或字号命令。但是在题名版面的源程序中出现换行控制符时, 另外加注的字体或字号命令会立即失效。所以假如你要打印若干行信息的话, 你得逐行添进变换字体或字号所必须的控制命令。

下面是一个控制题名部分版面的排版实例:

```
\documentstyle[12pt]{book}
.....
\title{Mathematical Methods for Physicists}
\thanks{Supported by a grant from the TJF} }
\author{Yi-feng Shen \thanks{To My Wife} }
\date{May 8, 1978}
.....
\begin{document}
\maketitle
.....
```

## 5.8 小页排版命令

所谓小页(minipage) 是指在常规的页面上模拟编排相对版心较小尺寸的独立的排版版面。例如在一些杂志或报刊上就经常出现这样的用于编排小块文章的小页版面。即使在科技论文中, 也少不了要编排插图图注或摘要等块状文字的小页版面。在 $\text{\LaTeX}$  排版系统中, 小页排版命令实际上只是将LR 排版盒子命令(只能对单行文字进行编排, 参见第3.7 节) 的基本思想在多行文字排版场合下作一些推广而已。

小页版面的控制可以通过两种排版命令方式来实现：一种是设置一个小页版面(minipage)的排版环境，而另一种则类似于LR排版盒子命令，建立一个可换行排版的小页排版盒子(parbox)。

### 5.8.1 小页排版环境命令

命令格式如下：

```
\begin{minipage}[对位参量]{小页版面宽度}
  <小页正文>
\end{minipage}
```

说明：

① 在行文状态下任何位置出现此命令时，系统就会将后续出现的处在小页排版环境中的正文内容视为一个与外界无关的独立排版盒子。这个盒子的宽度由小页版面宽度参数来确定，而高度则根据盒子内的信息量大小自动调节。

② 由于在行文状态下的任何位置上难于放置一个“巨型”字符——小页版面，系统给小页排版环境命令设计了一个对位参量。它的作用是对排版盒子相对前面的行文中心线作上下安置方式作注解。可选择的参量有两个：

t 此参量出现时，系统将盒子的左上角与前面的行文对齐；

b 此参量出现时，系统将盒子的左下角与前面的行文对齐。

如果用户省略此参量。那么系统将盒子纵向方向的中心线与前面行文的中心线拉齐。

③ 如果你在小页排版环境中的正文内容里使用了脚注排版命令，那么就可以将脚注的内容直接排在小页版面的底部。这种排版方式对于在图表类的版面底部添加些脚注文字很有用处。

例如小页排版环境程序实例：

```
...\fbox{See {\tt minipage} example:)}\
\begin{minipage}[t]{3cm} \it
This is an example of minipage environment
aligned on its top lines.
\end{minipage}
\ AND \
\begin{minipage}[t]{3cm}
One is also in the minipage environment\footnote{See
Page~\pageref{page:minipage}}
aligned on its top line.
\end{minipage}
```

可以得到：

... See minipage example: *This is an example of minipage environment aligned on its top lines.* AND One is also in the minipage environment<sup>a</sup> aligned on its top line.

<sup>a</sup>See Page 75

### 5.8.2 小页排版盒子命令

小页排版盒子命令的排版功能与用小页版面排版环境命令控制的相类似，只是命令的形式有所不同。小页排版环境命令是由“\begin”和“\end”配对，并加上注解参量“minipage”构成的。而小页排版盒子命令类似于无框线LR排版盒子命令“\makebox”，仅由一行命令构成，而将小页版面中的正文内容作为注解参量放在命令之中：

`\parbox[对位参量]{小页排版盒子宽度}{小页正文}`

说明：

① 在行文状态下任何位置出现此命令时，系统就会将命令中最后一对花括号中出现的小页正文内容视为一个与外界无关的独立排版盒子。这个盒子的宽度会根据参数小页排版盒子宽度来确定，而高度则由盒子内的信息量大小自动调节。

② 方括号内的对位参量的作用与在小页版面排版环境命令中的完全一样。如果用户省略此参量，那么系统将盒子纵向方向的中心线与前面行文的中心线拉齐。

③ 小页排版盒子命令中不能使用脚注排版命令。如果你在小页排版盒子里的正文内容中使用了脚注排版命令，那么你在小页排版盒子的底部将看不到脚注。

例如小样：

```
... \fbox{See {\tt $\backslash$parbox} example: } \
\parbox{3.5cm}{ \sl This is a simple example of
{\tt $\backslash$parbox}
command aligned on its \underline{center} lines. }
\ YOU WILL \
\parbox{3.5cm}{ see the second
{\tt $\backslash$parbox} example
also aligned on its \underline{center} line. }
```

可以得到：

... See \parbox example: *This is a simple example of \parbox command aligned on its center lines.* see the second \parbox YOU WILL example also aligned on its center line.

再如程序：

```
... {\bf See another one: } \
```

```

\parbox[b]{3.5cm}{\sf This is another example of
{\tt $\backslash$parbox}
command aligned on its {\it bottom} lines.}
\ \fbox{\Large AND THIS} \
\parbox[t]{3.5cm}{One is also the
{\tt $\backslash$parbox}
example aligned on its {\em top} line.}

```

可以输出:

This is another exam-  
ple of `\parbox` com-  
mand aligned on its bot-  
... See another one: *tom* lines.

AND THIS

One is also  
the `\parbox` example  
aligned on its *top* line.

## 5.9 摘要排版控制命令

摘要(Abstract, 也叫做提要) 是科技论文中不可缺少的一部分。通常排在论文题名页或文章的第一面上(位于文章题名部分与正文之间)。在文章类型(article) 或是报告类(report) 文件的文件格式中,  $\text{\LaTeX}$  是通过设置摘要排版环境来编排摘要的, 其控制命令为

```

\begin{abstract}
  <摘要正文>
\end{abstract}

```

说明:

① 如果用户没有设置题名页, 则在文章类型(article) 的文件格式中的摘要会被排在命令所处的位置; 而对于报告类(report) 文件或是设置了题名页版面的文章类文件, 则摘要会被另面排版。执行排版时正文部分将以引文版式排印。在摘要正文部分上面, 系统会自动为你居中打印上黑体字“Abstract”, 而摘要正文部分的西文字以相对基准字体小一号的字体排版。

② 由于书籍一般没有摘要, 因此摘要排版环境命令不能在书籍类(book) 文件中使用。

## 5.10 分栏排版控制命令

如果没有特别的分栏排版命令控制版面, 根据 $\text{\LaTeX}$  约定, 系统总是以通栏(单栏) 样式来排版。假如用户希望所编排的文件版面全部按照通常学术刊物那样分成两栏排

版, 则只要在源文件最前面的文件格式命令之中添进双栏版式注解参量即可。例如:

```
\documentstyle[12pt,twocolumn]{article}
```

但有时用户希望从某一页起开始按双栏版式编排文件(例如名词或作者索引版面), 则须使用如下的双栏排版控制命令:

```
\twocolumn
```

说明:

① 系统在源文件中读到此命令时, 即将排版方式切换到双栏版式状态, 并将命令以后的正文内容以双栏形式另页起排。

② 如果用户从某一页起又要求恢复通栏(单栏)版式, 那么可输入单栏版式控制命令:

```
\onecolumn
```

这时, 系统又会将后续内容另面以通栏版式起排。

③ 注意系统没有提供多栏版式( $n > 2$ )的排版控制命令。

## 5.11 版式全局说明控制命令

在文件格式命令“\documentstyle”与正文文稿排版环境起始命令“\begin{document}”之间是 $\text{\LaTeX}$ 源文件的版面排版样式的全局说明区域。在这一区域内, 可以设置一些专用控制命令, 用以对正文版心的宽度、高度、行距、天头、脚注和图形状态中的基本长度单位等作具体注解。

### 5.11.1 版心尺寸控制命令

版心(见图5.1)是指每个版面上包括脚注在内的正文部分内容(不含页眉或页码)。决定版心尺寸的参数主要有两个: 版心宽度和高度。假如你需要将排版后的输出文件按出版社的要求设置版心尺寸, 那么可使用如下两条命令:

```
\textwidth=〈版心宽度〉
```

```
\textheight=〈版心高度〉
```

这里的版心高度不包括页眉部分。版心宽度和版心高度的长度单位可选取“mm”、“cm”、“pt”、“in”和“em”等(输入命令时尖括号“〈〉”不要键入, 以下同)。例如, 要设计一个在中国出版界常用的大32开版本的正文版心(尺寸为105mm×155mm), 你可在全局说明区域输入下两条命令:

```
\textwidth=10.5cm
```

```
\textheight=15.5cm
```

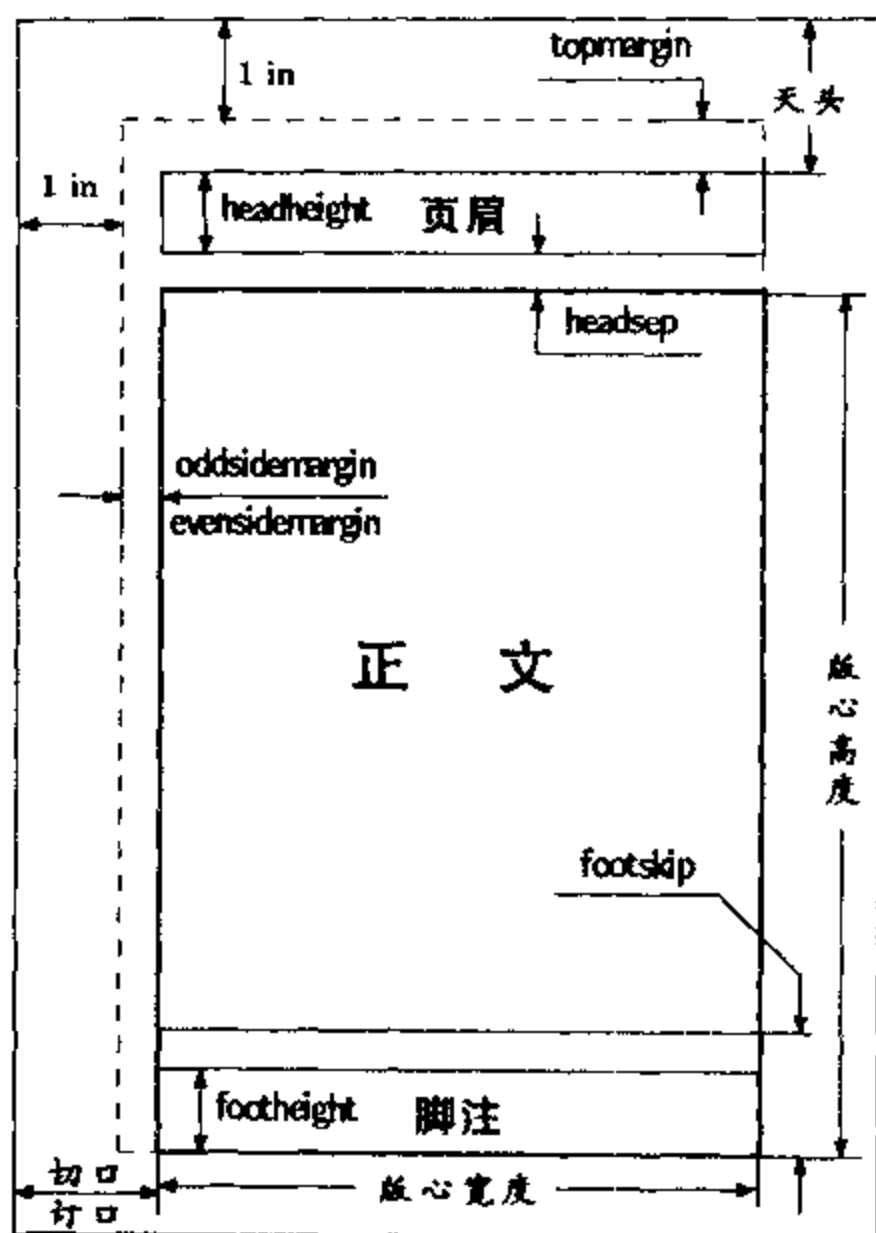


图 5.1 A4 纸版式全局说明示意图

### 5.11.2 页边距控制命令

白边(margin, 也叫页面空白边)是指除版心(见图5.1)之外,在页面(如A4纸)纸边界四周的空隙部分。 $\text{\LaTeX}$ 系统中,版面的页边空白边距的控制只涉及纵向顶端和横向左侧两个区域。通常印刷物有单面和双面两种。对于单面印刷的版面,只须考虑天头尺寸和订口尺寸就行了。然而对于双面印刷,就要对奇数、偶数页的订口尺寸和切口尺寸分别加以考虑了。

① 天头尺寸 天头尺寸特指页面纸的顶端到文章的页眉(没有页眉时为文章正文的顶部)之间的距离。 $\text{\LaTeX}$ 控制天头尺寸时,以距页面纸顶部边界1英寸(2.54cm)的位置(见图5.1中的虚线)作为新的边界线起始线。所以如果你选定了实际的天头尺寸,那么可以使用如下命令来间接地控制:

$\backslash\text{topmargin} = \langle \text{天头尺寸} - 1 \text{ 英寸} \rangle$

当“ $\backslash\text{topmargin}$ ”赋值为零时,页眉顶部边界即与图5.1中上方的横虚线重合(虚线距A4纸边界1英寸)。由于以虚线处作为边界线起始线,命令“ $\backslash\text{topmargin}$ ”所赋之值可以为正负实数(负数绝对值不能大于1英寸)。例如,一篇文章的版心高度数值比较大,希望

天头部分仅留下1.5cm的页边距, 那么应在全局说明区域内输入命令

```
\topmargin = -1.04 cm
```

② 奇数页订口尺寸 订口尺寸是指书刊页面纸的装订线一侧边缘到文章的版心边界之间的距离(见图5.1的左侧)。由于 $\text{\LaTeX}$ 在排版时只考虑横向页边距的左侧。所以当单面印刷时, 只须考虑订口尺寸(页码的奇数或偶数没有什么差别); 如果是双面印刷, 则订口尺寸的控制仅对奇数页有效。所以订口尺寸只须考虑奇数页就行了。与天头区域边界起始点相似,  $\text{\LaTeX}$ 亦将距离页面纸左边界1英寸的位置定义为新的起始点。所以当你确定了实际的奇数页的订口尺寸时, 可以在全局说明区域内使用命令

```
\oddsidemargin = <实际奇数页订口尺寸-1英寸>
```

③ 偶数页切口尺寸 切口尺寸是指书刊页面距装订线另一侧边缘(书刊的每一页纸的外侧)到文章的版心边界之间的距离。当单面印刷时, 无须考虑切口尺寸; 如果是双面印刷, 则切口尺寸的控制会对偶数页起控制作用(对于偶数页版面, 切口处一侧处于版面的左侧, 所以 $\text{\LaTeX}$ 在排版时会考虑如何调节切口尺寸)。以新的起始点作为边界的偶数页切口尺寸的控制命令为:

```
\evensidemargin = <实际偶数页切口尺寸-1英寸>
```

考虑一个在A4纸上进行双面印刷的实际排版例子: 设文件格式为报告类, 基准字号为11pt; 假定版面的版心尺寸(宽度 $\times$ 高度)为155mm $\times$ 225mm; 订口尺寸要比切口多一厘米留作装订用, 且剩余页面空间要保证版心左右居中; 天头部分须留出2cm宽的实际页边界。由于A4纸总宽度为210mm, 扣除版心宽度(155mm), 并留出10mm作为页面装订部分, 最后还剩45mm。平均分配到版心的两侧, 则实际偶数页切口尺寸为22.5mm, 而实际的奇数页订口尺寸为32.5mm。根据以距离页面纸边界25.4mm(1英寸)的位置作为新的边界线的原则, 你可以采用如下全局说明命令:

```
\documentstyle[11pt,twoside]{report}
\textwidth = 155 mm
\textheight= 225 mm
\topmargin = -5.4 mm
\oddsidemargin = 7.1 mm
\evensidemargin = -2.9 mm
.....
\begin{document}
.....
```

以上全局说明命令的先后秩序没有规定, 各条命令也不是一定要分行输入。

### 5.11.3 页眉尺寸控制命令

决定页眉部分版面尺寸的几何参数只有两个: 页眉的高度(headheight) 和将页眉

底部边界与版心顶部边界分开的距离(headsep.)。控制这两个参数的控制命令格式分别为:

```
\headheight = <页眉高度>
\headsep = <页眉与版心的间距>
```

#### 5.11.4 脚注尺寸控制命令

确定脚注部分版面尺寸的参数也有两个: 脚注的高度(footheight) 和脚注底部边界与版心底部边界之间距离(footskip)(见图5.1)。两者的控制命令格式分别为:

```
\footheight = <脚注高度>
\footskip = <脚注底至版心底的距离>
```

注意:

用户在一般情况下不要自行设置脚注高度, 因为系统会根据脚注正文信息量的大小自动调节脚注版面的高度。

#### 5.11.5 旁注尺寸控制命令

确定旁注部分版面尺寸的参数有三个: 旁注的宽度(marginparwidth)、旁注离版心边界的间距(marginparsep) 和上下两近邻旁注之间的最小距离(marginparpush)。相应的控制命令格式分别为:

```
\marginparwidth = <旁注宽度>
\marginparsep = <旁注至版心距离>
\marginparpush = <上下旁注最小间距>
```

注意:

① 除非特殊需要, 一般情况下用户不要自行设置上述各参数, 因为系统会为用户自动设置三个比较合适的参数。

② <上下旁注最小间距> 参数只是在由正文内容决定的两相邻旁注版面之间发生交叠时才生效, 其数值用于控制上下两旁注间须错开的最小距离。

#### 5.11.6 行距控制命令

在全局排版说明区域内, 行距的控制命令格式为

```
\renewcommand{\baselinestretch}{相对行距}
```

这里相对行距 参数是一个无单位的正实数。相对行距 数值的意义类似标准机械式打字机上的行距控制键调节时所对应的数值。例如当取值为“1.5”时即为通常出版社对作者所要求的隔行打字的排版效果。这种打印方式输出的效果看上去虽不美观, 但是易于编辑和修改。在标准情况下, 数值应取“1”: 如果用户未在全局排版说明区域内设置



行距控制命令，系统将自动设置为“1”。当然，用户还可以取其它任意实数，例如可取“1.8”或“2”。

说明：

① 行距命令只是对行与行之间的空距作相对控制。换句话说，对于不同字号的行文，行距的尺度是不同的。

② 以上命令只能在全局排版说明区域内使用。如果在正文中的某个版面内需要作绝对的行距调整，则可使用以下的文中行距控制命令：

`\baselineskip = <绝对行距>`

这里绝对行距 必须带有单位，其数值大小直接定义了相邻两行底线之间距离的绝对值，且不会因字体的变换而作相对变化。

### 5.11.7 段落间距控制命令

在西文排版过程中，每两个相邻的段落之间常常需要在垂直方向上留出一些“额外”的空白间距。所谓“额外”空距，是指除了像正常的两行行文之间留有的距离之外，还需要附加的距离。L<sup>A</sup>T<sub>E</sub>X 系统根据省略约定，已经替用户考虑了如何设置段落间距。但是如果用户根据特殊版面的要求，需要自己设置段落间距的话，那么可在全局排版说明区域内使用如下控制命令

`\parskip = <段落绝对间距>`

其中段落绝对间距 必须带有单位，它的数值大小定义了另起一个段落之前在纵向留出的空白间距的绝对值。在正文中，仍然可使用此命令作段落间距的更换。

### 5.11.8 段首缩排控制命令

如同中文段落第一行起始位置需要空出两个中文字间隔一样，西文段落的最前面也可根据版面的要求缩进一些空白间距排版，通常简称之为缩排。与以方块字为主体的中文所不同的是，西文每个字母的宽度并不一样，因此不能简单用若干个字母宽度来确定段首的缩进间距。L<sup>A</sup>T<sub>E</sub>X 系统的段首缩进排版间距能连续可调，控制命令的格式为：

`\parindent = <段首缩进宽度>`

段首缩进宽度 参量选取时应带有L<sup>A</sup>T<sub>E</sub>X 系统允许使用的长度单位，它确定了每个自然段段首第一个单词前面留出空间的大小。例如在段首需要空出0.8 厘米时，可由命令

`\parindent = 0.8cm`

来控制。如果用户不希望正文中的某一段起始位置缩进排版，则可在段首插入顶格排版命令“`\noindent`”；如果用户不打算让所有的段落起始位置留有任何空间，则可在全局排版说明区域内输入如下命令：

`\parindent = 0pt`

段首缩进间距控制命令可以在正文中的任意位置使用。如果用户希望在某些通过强迫换行后顶格排版的行首处亦留出像段首缩进排版时一样的横向空距(即齐肩排版), 则可在该行的起始位置插入行首缩排命令

`\indent`

### 5.11.9 栏间空距排版命令

假定正文版面按照用户要求按照双栏版式排版, 则系统会自动在两栏边界之间留出一些空白间距。如果系统编排的栏间距不能满足你的特殊要求, 则可使用连续可调的栏间空距排版命令:

`\columnsep = <栏间距>`

为了使版面看上去美观大方, 系统还提供了能在两栏之间画一竖线的排版控制命令

`\columnseprule = <线径尺度>`

以上长度单位可任取 $\text{\LaTeX}$ 合法单位。

## 5.12 版面撑满控制命令

所谓撑满(fill), 是指在排版过程中, 根据文稿的具体内容, 将行文在版面上沿横向或纵向朝当前排版环境所限制的版心边界均匀地拉开。 $\text{\LaTeX}$ 为用户提供了实现横向和纵向两种类型版面撑满的控制命令。用户可在源文件的适当位置插入若干控制命令, 从而达到使文件的正文部分将版面左右或上下的边界撑足。

### 5.12.1 横向撑满控制命令

$\text{\LaTeX}$ 系统中提供的横向撑满控制命令格式为:

`\hspace{\fill}`

说明:

① 在某一孤立行中的任意位置插入此命令后, 可以将控制命令两边的行文内容按照版面宽度向左右边界撑足排版。此命令的设置实际上相当于插入一个自动可调的动态横向空距命令。

② 在同一行中可以在不同的位置插入若干个横向撑足命令, 执行排版时,  $\text{\LaTeX}$ 会将行文内容分成若干段等间隔地均匀拉开。

③ 由命令控制的向左右边界撑足的实际宽度将视当前设定的排版环境所限制的版心边界而定。例如, 在小页排版环境, 或引文排版环境中, 边界线将由这些特殊排

版环境确定。此外，段首的缩进空距命令或其它专门设置的排版控制命令的功能也不受影响。

④ 此命令有一个缩写式“\hfill”，其功能与原式完全等价。

例如源程序

```
\parbox{10cm}{This is an example of \hspace{\fill} a stretched space.\\
               Here are two \hfill examples of \hfill stretched spaces.\\
               And here \hfill are three \hfill equal \hfill ones.}
```

可以获得如下横向撑满的排版效果：

```

This is an example of                a stretched space.
Here are two                examples of                stretched spaces.
And here                are three                equal                ones.
```

### 5.12.2 纵向撑满控制命令

纵向撑满控制命令的标准格式为：

```
\vspace{\fill}
```

说明：

① 在某一页中的任意行位置插入此命令后，执行排版时就具有类似横向撑满控制命令的功能。它可以控制排版系统将命令前后各行的行文内容按照版面高度向上下边界撑足撑版。

② 在同一页中可以在不同的行位置插入若干个纵向撑足命令，执行排版时， $\text{\LaTeX}$ 会将行文内容分成上下若干段等间隔地均匀拉开。

③ 这个命令也有一个缩写式“\vfill”，它的功能与原式相同。

### 5.12.3 点线撑满控制命令

点或线撑满控制命令的功能与横向撑满控制命令非常接近，只是在排版时除了将行文向面边撑足之外，还用离散的点或连续的水平线将空白同距处填满。点撑满的控制命令形式是：

```
\dotfill
```

这个命令常被用于编排目录之类的文件。水平线撑满的控制命令形式是：

```
\hrulefill
```

在同一行中，点或线的撑满控制命令与横向撑满控制命令可以混合使用，撑版时各个命令所控制的空距长度相等。例如由撑程序：

```
1.1 Examples with \verb|\dotfill| command \dotfill\ (28)\
```

```
.....
```

---

3.2 Examples: \hfill \verb|\hrulefill|  
 \hrulefill\ command \dotfill\ (50)

可以获得如下版面撑满的排版效果:

1.1 Examples with \dotfill command ..... (28)  
 :  
 3.2 Examples: \hrulefill \_\_\_\_\_ command ..... (50)

## 6

# 数学公式排版命令

本章详细介绍如何使用 $\text{\LaTeX}$ 排版命令来进行各种复杂的数学公式的编排。例如怎样打印上下标、特殊数学符号、常用的数学函数、分式、根式以及矩阵等。

## 6.1 数学排版概述

许多科技文章总会引进一些数学公式，例如一元二次方程标准式 $ax^2 + bx + c = 0$ 的解可表示为

$$x_{\pm} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

显然，用古老的传统机械式打字机是无法打印上述在数学方程中还不算是太复杂的公式的；即使装备了比较齐全的特殊数学符号的铅字，仍然难以将上述公式打印出令人感到满意的形式。如果请排字工人编排，亦非是件易事。自从有了计算机轻印刷系统，数学方程的排版才日臻完美。不过，国内现有的汉字排版系统对数学公式的排版均须受专门化训练，一般人较难掌握。究其原因大抵是排版用的命令太复杂，控制符号太古怪，因而可读性较差，使人难以掌握。 $\text{\LaTeX}$ 排版系统在这方面却显示了巨大的威力。它将编排数学公式变得非常容易，一个非专业化的工作人员可以很快地掌握数学公式排版技能。 $\text{\LaTeX}$ 之所以深受广大科技人员的欢迎，这是其中的重要原因之一。

$\text{\LaTeX}$ 对于数学公式的排版是采用批处理的方法，即将数学符号和排版命令同时输入，由系统对程序自动执行排版。数学排版命令可以相互嵌套，灵活使用，以解决非常复杂的数学方程的排版。

数学公式的排版方式与一般的西文行排状态是不同的。例如，在数学公式中，所有的常规西文字母都采用意大利斜体字，每个字母有其自身特定的数学意义，在排版时不会出现像行文中那样的几个特殊字母的联体化。为了与一般西文正文的行排状态相区别， $\text{\LaTeX}$ 专门单独设计了一种编排数学公式的状态模式。在这个特殊状态中， $\text{\LaTeX}$ 系统的部分排版规则和控制命令的功能将发生一些变化。所以，用户如要排数学

公式首先必须要实现由一般行文状态到数学状态的切换。

对于一般西文文稿的行文中出现的单个数学公式(包括简单数学符号和函数), 或者以单独一行形式表述的较复杂的数学公式, 用户可以使用下面介绍的任意一种环境命令来实现从行文排版状态到数学排版状态的转换。

### 6.1.1 行中数学公式状态命令

所谓行中数学公式, 是指行文排至某一行时, 正文中间不换行而直接引用了数学符号、简单函数或方程。例如前面提到的一元二次方程的数学表达式就是被排在行文之中。进入和退出行中数学公式状态的控制命令是

```
\begin{math}
  <数学公式>
\end{math}
```

当然由于单个的换行标记对 $\text{\LaTeX}$ 系统只等价于一个空格。所以, 在数学公式内容很少的情况下, 你完全可将上面命令合成一行来输入, 即可将命令任意插在正文中与数学公式前后的行文写在一起,

```
... <行文> ... \begin{math} <数学公式> \end{math} ... <行文> ...
```

然而, 行中数学公式状态命令的使用对于科技论文来说是非常频繁的, 因此, 如果只是为了几个很短的数学式子, 或者单个变量而多次重复输入

```
\begin{math} ..... \end{math}
```

就会令人感到异常乏味。所以 $\text{\LaTeX}$ 为用户设计与上面标准形式完全等价的另外两种适合于编排数学短式子的简单命令形式。

简式1:  $\backslash( <数学公式> \backslash)$

简式2:  $\$ <数学公式> \$$

以下是一个使用三种形式的行中数学公式状态命令的排版实例。

```
The mathematical formula \begin{math} x' + y'' =
z''' < 2w \end{math} is now easy to type. \LaTeX{}
regards the formula \backslash(a > 2b \backslash) as a word. Remember
a single variable $x$ is also a formula.
```

输出结果:

The mathematical formula  $x' + y'' = z''' < 2w$  is now easy to type.  $\text{\LaTeX}$  regards the formula  $a > 2b$  as a word. Remember a single variable  $x$  is also a formula.

也许读者现在已经体会到为什么我们以前总是要提醒用户慎用控制字符。在 $\text{\LaTeX}$ 源程序里, 一个简单的美元符号“\$”或控制符“\”就能立即将当前的排版状态切换到数学公式排版状态之中, 而后续的第二个“\$”符号或“\”又会使系统迅速退回到行文排版状态之中。简式2 由于最简单, 所以使用得也最广泛, 不过也容易出错。因为前后(进入与退出)符号使用了相同的字符, 容易疏忽和遗漏。在运行系统执行排版时最多的出错信息就是使用者忘了另一个用来命令退出数学公式排版状态的美元符号“\$”。

作为排版基本知识, 建议用户不要在任何一段行文的起始位置使用行中数学公式状态排版命令(即不要以数学公式来引出一段行文)。

### 6.1.2 独立数学公式状态命令

独立数学公式是指正文中所引用的另起一行表述的单个数学方程式, 像前面所引用的一元二次方程之解的数学表达式的排版方式就属这一类。进入和退出独立数学公式状态的控制命令是

```
\begin{displaymath}
  <数学公式>
\end{displaymath}
```

这个控制命令同样也有两个简单的等价表达式:

简式1: `\[ <数学公式> \]`

简式2: `$$ <数学公式> $$`

例如我们可将上面例子中的命令改成独立数学公式状态命令的形式:

```
The mathematical formula \begin{displaymath} x' + y'' =
z''' < 2w \end{displaymath} is now easy to type. \LaTeX{}
regards \[ a > 2b \] as a displayed formula.
```

输出结果:

The mathematical formula

$$x' + y'' = z''' < 2w$$

is now easy to type.  $\text{\LaTeX}$  regards

$$a > 2b$$

as a displayed formula.

### 6.1.3 独立编号数学方程命令

本命令与独立数学公式状态命令的功能基本相同, 唯一的差别是系统会自动为所编排行的单个数学方程式编号。一般情况下(除非设置其它方式的排版命令), 编号是加

括号居右排版的。独立编号的数学方程控制命令的标准格式为

```
\begin{equation}
  <数学方程式>
\end{equation}
```

注意这个命令格式没有等价的简单表达式。系统在执行排版的时候, 会对文件内所有在独立编号数学方程命令环境中编辑的单个式子依次编号。所以用户在对文中所引用的方程作增删时无须担心方程的编号问题。下面是一个数学公式状态中无方程编号与有编号的对比排版实例:

Here is an example of an unnumbered displayed equation:

```
\begin{displaymath} 0.3x + y/2 = 4z \end{displaymath}
```

and here the same equation numbered:

```
\begin{equation} 0.3x + y/2 = 4z \end{equation}
```

输出结果

Here is an example of an unnumbered displayed equation:

$$0.3x + y/2 = 4z$$

and here the same equation numbered:

$$0.3x + y/2 = 4z \quad (6.1)$$

说明:

① 如果用户没有在文件格式命令中对方程的排版方式加以注解, 则方程总会居中排版, 且序号总是置于方程的右侧, 并与版心边界拉齐。

② 假如用户希望改变方程的排版方式(例如将所有方程式向左侧对齐, 或将方程编号排在方程左侧), 则可在文件格式命令中选择如下版式注解参量:

`leqno` 将方程序号置于方程式左侧。

`fleqn` 所有方程式靠左侧拉齐排版; 每个方程式离版心左边界有一固定宽度。

当用户设置了方程式靠左侧拉齐排版的版式注解参量时, 用户还可以使用下面的数学公式左边距控制命令

```
\mathindent = <左边距>
```

来改变方程式离版心左边界的空距。

## 6.2 数学公式状态下的基本规则

与行文排版状态相比,  $\text{\LaTeX}$  在数学公式状态下执行排版操作时有许多不同之处。下面介绍一些数学公式状态下的基本排版规则。



### 6.2.1 字 距

进入数学公式的排版环境后, 系统将忽略源文件中所有以任何形式键入的ASCII空格符, 包括在输入文件过程中另起一行时所击的计算机键盘上的回车键(硬回车), 以及用于产生多个空格的Tab键信息。 $\text{\LaTeX}$ 在读进用户输入的数学公式排版命令之后, 会自动按照书刊印刷的一般规则以最佳方式来编排数学公式。系统会根据排版需要自动调节各个数学符号之间的距离。

在数学公式的排版环境里, 输入的任何西文的行义句子只会被视为一个冗长的数学公式(单词之间的空格已经无效), 试比较下列排版程序中行文排版状态和数学公式排版状态的区别:

```
This sentence is an example of \LaTeX usual operation mode.\\
\begin{math} This line is an example of mathematical formula.
\end{math}
```

输出结果:

This sentence is an example of  $\text{\LaTeX}$  usual operation mode.  
 This line is an example of mathematical formula.

如果你在输出时觉得部分数学符号之间显得太拥挤或者太稀疏, 那么可使用字距控制符来调整间距。例如你可采用控制符“\,”(另外插入较小的字距)、“\!”(后退较小的字距)或其它合法的字间距控制命令在数学公式的排版环境中,  $\text{\LaTeX}$  提供了如表6.1所列出的几种字距控制命令。不过在使用这些字距命令前最好先去研究一下其它标准数学印刷物中是如何编排数学公式的, 否则的话, 额外添加的字间距也许会使别人难以理解你给出的数学公式的意义。所以一般情况下, 不要在数学公式的排版环境中使用字间距控制命令。

表 6.1 数学公式状态字距命令表

控制命令	意义	字距(em)	排版效果
\,	行文标准词间距	0.33	<i>x x</i>
\;	数学式宽字距	0.30	<i>x x</i>
\:	中等字距	0.23	<i>x x</i>
\.	窄字距	0.17	<i>x x</i>
\(省略)	无字距	0	<i>xx</i>
\!	负窄字距	-0.17	<i>xx</i>

说明:

① 表中的字距单位“em”是以字母“M”的宽度为基本单位。对于标准的10点(pt)正文字体, 宽度值大约为3.5mm。

② 最后一行中的负窄字距命令相当于后退一点距离, 一般很少使用。

③ 表中的字距命令“\\_”和“\,”也可以在通常的行文排版状态中使用, 但是其它指令只能在数学公式状态下使用。

### 6.2.2 字体

在数学公式状态下, 所有输入的未加字体转换命令注解的英文字母均以意大利斜体字形出现(这是目前所有出版社采用的排版方式)。由于每个字母被系统视为一个有特定意义的数学符号, 所以在排版时字母之间会空出一点额外的字距, 使得你在输出时看上去与行文状态下的由意大利字体排版的词句有些不同。在数学公式排版环境中, 仍然可用字体转换命令来变换字体, 但是不要企望得到与行文排版状态下印出的相同形式, 因为字词间的空格均被忽略了。

### 6.2.3 数学符号

有许多特殊的数学符号和字体可在数学公式排版环境中用控制命令印出。例如希腊字母、书写体以及许许多多的复杂的数学运算符号。值得注意的是, 这些符号不能在一般行文状态下直接使用。行文中出现这种控制命令将视为非法。关于这部分内容, 我们将在后面专门安排一节作详细介绍。

### 6.2.4 公式的分行和行距

数学公式排版环境中不允许分段记号(一行空格或连续几个硬回车字符, 或控制命令“\par”)的出现。因为通常在数学公式中很少出现另起一段行文的情形。如果你在数学公式排版环境中输入了空行记号, 那将会被系统视为你想退出数学公式排版状态, 于是会在终端上显示出错信息。如果你实在想要把公式拆成上下空开一定距离的形式, 可使用换行控制符外加一个纵向行距控制命令, 或者采用本章后面向你介绍的多行数学公式的排版方法。

## 6.3 上下标命令

数学公式中用的最多的恐怕就数变量的上标和下标了。正因为如此,  $\text{\LaTeX}$  将上下标命令设计得异常简单。设计的最重要的思想是由简单的控制符将上标或下标的内容定义在一个盒子里(一个独立的数学排版环境)。这样在盒子里可继续编排包括上下标命令在内的数学公式, 如此可相互嵌套, 使用非常方便。

### 6.3.1 上标排版命令

类似于用户比较熟悉的 Basic 程序语言,  $\text{\LaTeX}$  的上标排版控制命令仅由一个 ASCII 尖角控制字符“^”和一对带参量的花括号“{}”组成。具体格式为

$\wedge$ {上标数学表达式}

这里字符“ $\wedge$ ”作为上标的排版控制命令，而后面的花括号“{}”则是一个对外界无影响的排版小盒子，它构成一个独立的数学公式排版环境。

例如幂函数 $x^2$ 和 $y^a$ 的排版可由上标排版命令“ $\$x\wedge\{2\}\$$ ”和“ $\backslash(y\wedge\{a\})$ ”来分别控制。如果上标中还有幂函数，即所谓上标的上标，则可在上标的独立环境中继续使用上标排版命令。例如像下面这样的复杂方程式，

$$x^{y^z} = (1 + e^x)^{-2zy^w}$$

可以由控制程序

$\backslash[ x\wedge\{y\wedge\{z\wedge\{w\}\}\}=(1+\{\rm e\}^{\{x\}})^{\{-2zy\wedge\{w\}\}} \backslash]$

来实现排版。

### 6.3.2 下标排版命令

下标排版控制命令同样很简单。与上标命令不同的地方是尖角控制字符由下划线(底线)字符“ $\_$ ”取而代之。它的标准格式为

$\_$ {下标数学表达式}

注意这里花括号“{}”仍然构成了一个独立的数学公式排版环境。如果你要打印线性代数中的矩阵元表式 $a_{ij}$ ，则可通过输入命令“ $\$a_{\{ij\}}\$$ ”来完成。

注意：

① 由于上面介绍的上下标命令中的花括号内依然是处于数学公式模式，系统仍将其中的字体取为意大利斜体字。如果遇到上下标要用罗马字体的情形，用户可利用字体控制命令“ $\rm$ ”来切换到罗马字体模式。在行文中，某些特殊的西文单词右上端或右下方也会需要加些上标或下标注解词。如果要处理这种情况，切记只有数学公式状态才能排上下标。熟练地使用模式切换字符“ $\$$ ”，可帮助你实现许多复杂版的排版。例如你要排一个带上下标的艺术化单词

$\text{Super}_{\text{script}}$

你可在字母“Su”的后面先将行文排版状态切换到数学公式排版状态中，然后再使用上下标命令。所以你可输入

$\text{Su}\$ \{ \rm per \} \_ \{ \rm b \} \$script$

② 一般情况下，系统会自动将不同级别的上下标字号逐级变大(即字型逐级缩小)。如果你不满意，可选择使用字号转换命令来放大或缩小上下标的字型。

③ 如果在一个数符号后面需要同时排上下标(例如像上面的艺术化单词)，则输入上下标排版命令的先后顺序对输出时的打印效果无任何影响。因此，你还可将上面的

例子中的排版程序改成

```
Su$_{\rm b}^{\rm per}$script pa
```

之后,照样能获得与上面相同的排版结果。

④ 数学公式环境中,对于源文件中出现的独立上下标命令,系统将按照上面列举的排版实例形式排版。但是假如上下标命令前面出现一些特殊的可变型数学符号,如积分号和求和号(参见本章第四节中的第四小节),则上下标的位置会根据排版环境的不同而变化。例如求和号的上下限可能会被放在求和号的顶部和底部。这是根据数学印刷物的标准而精心设计的,你无需为此操心。

## 6.4 数学符号排版命令

数学公式实际上是由一些特定形式的变量(特殊字母)和数学运算符号有规则的堆砌而成的。 $\text{\LaTeX}$ 系统装备了相当齐全的可供排版的各种特殊规格的数学公式用的字体和运算符号字库。输出这些数学符号是由相应的排版命令控制完成的。当然这些符号只有在数学公式排版状态下方能使用。下面分别介绍一些系统提供的数学符号的排版命令。

### 6.4.1 希腊字母

#### 一、小写希腊字母

常用的希腊字母(Greek letter)的小写形式是由各个字母的小写英文名字之前加上“\”来获得的(见表6.2的上面一部分)。提请读者注意的是小写希腊字母中一些字母有变形式(均附在表6.2中),这些字母的控制命令只不过是从小写希腊字母命令中的英文名字前加上了一个前缀“var”(变形)而已;此外希腊小写字母“o”直接采用意大利字体(数学公式状态中的一般英文字体)。

#### 二、大写希腊字母

大写希腊字母的打印并没有通用的规则可循。这倒并不是因为打印这些字母比较困难,而是由于许多希腊字母的大写形式与罗马字体完全一样。因而在排这些字母时你可直接取自罗马字体中相应的字母。例如希腊字母“ $\alpha$ ”和“ $\beta$ ”的小写形式可由控制命令“\alpha”与“\beta”来分别产生,而大写形式则是由“\rm A”和“\rm B”来控制打印的。至于那些在26个大写英文字母中找不到的希腊大写字母,则只要把那些小写希腊字母命令中的英文名字的第一个字母换成以大写开头就可以了(详细指令可见表6.2的下面部分)。

表 6.2 希腊字母控制命令表

小 写 希 腊 字 母					
字母	控制命令	字母	控制命令	字母	控制命令
$\alpha$	<code>\alpha</code>	$\iota$	<code>\iota</code>	$\varrho$	<code>\varrho</code>
$\beta$	<code>\beta</code>	$\kappa$	<code>\kappa</code>	$\sigma$	<code>\sigma</code>
$\gamma$	<code>\gamma</code>	$\lambda$	<code>\lambda</code>	$\varsigma$	<code>\varsigma</code>
$\delta$	<code>\delta</code>	$\mu$	<code>\mu</code>	$\tau$	<code>\tau</code>
$\epsilon$	<code>\epsilon</code>	$\nu$	<code>\nu</code>	$\upsilon$	<code>\upsilon</code>
$\varepsilon$	<code>\varepsilon</code>	$\xi$	<code>\xi</code>	$\phi$	<code>\phi</code>
$\zeta$	<code>\zeta</code>	$\omicron$	<code>\omicron</code>	$\varphi$	<code>\varphi</code>
$\eta$	<code>\eta</code>	$\pi$	<code>\pi</code>	$\chi$	<code>\chi</code>
$\theta$	<code>\theta</code>	$\varpi$	<code>\varpi</code>	$\psi$	<code>\psi</code>
$\vartheta$	<code>\vartheta</code>	$\rho$	<code>\rho</code>	$\omega$	<code>\omega</code>

大 写 希 腊 字 母					
字母	控制命令	字母	控制命令	字母	控制命令
$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>
$\Psi$	<code>\Psi</code>	$\Omega$	<code>\Omega</code>		

### 6.4.2 书写体

L<sup>A</sup>T<sub>E</sub>X 排版系统提供了26个大写英文字母的书写体(calligraphic letter)形式:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

这些字母是在数学公式排版状态下, 由字体转换命令“`\cal`”来控制打印的。例如你在文中想打印一段文字

Choose a function  $\mathcal{F}$  with  $\mathcal{F}(x) > 0$

你可在源文件里输入

Choose a function `\cal F` with `\( {\cal F} (x) > 0 \)`

读者不妨思考一下在这个例子中第一次出现的命令“ $\mathcal{F}$ ”中为什么没有加花括号而后面出现的却一定要加。另外请注意, 一般系统没有配备英文字母的书写体小写形式。

### 6.4.3 特殊数学符号

L<sup>A</sup>T<sub>E</sub>X 排版系统为用户提供了大量的特殊数学符号。这当中的很少一部分, 例如像以下这些运算字符

$$+ \quad - \quad / \quad = \quad < \quad > \quad ( \quad ) \quad [ \quad ]$$

就可以直接由键盘上对应的字符输入。但是另外绝大部分符号则要靠控制命令来获得。表6.3至6.6分别列出了数学运算符、逻辑运算符、箭头符号，以及各种类型的花样符号表。

表 6.3 数学运算符表

运算符	控制命令	运算符	控制命令	运算符	控制命令
$\pm$	<code>\pm</code>	$\dagger$	<code>\dagger</code>	$\bigcirc$	<code>\bigcirc</code>
$\mp$	<code>\mp</code>	$\ddagger$	<code>\ddagger</code>	$\oplus$	<code>\oplus</code>
$\times$	<code>\times</code>	$\amalg$	<code>\amalg</code>	$\ominus$	<code>\ominus</code>
$\div$	<code>\div</code>	$\cap$	<code>\cap</code>	$\otimes$	<code>\otimes</code>
$\cdot$	<code>\cdot</code>	$\cup$	<code>\cup</code>	$\oslash$	<code>\oslash</code>
$\bullet$	<code>\bullet</code>	$\uplus$	<code>\uplus</code>	$\odot$	<code>\odot</code>
$\circ$	<code>\circ</code>	$\sqcap$	<code>\sqcap</code>	$\triangleleft$	<code>\triangleleft</code>
$*$	<code>\ast</code>	$\sqcup$	<code>\sqcup</code>	$\triangleright$	<code>\triangleright</code>
$\star$	<code>\star</code>	$\wedge$	<code>\wedge</code>	$\triangleleft$	<code>\lhd</code>
$\diamond$	<code>\diamond</code>	$\vee$	<code>\vee</code>	$\triangleright$	<code>\rhd</code>
$\setminus$	<code>\setminus</code>	$\triangle$	<code>\triangle</code>	$\triangleleft$	<code>\unlhd</code>
$\wr$	<code>\wr</code>	$\nabla$	<code>\nabla</code>	$\triangleright$	<code>\unrhd</code>

表 6.4 确定型逻辑运算符表

运算符	控制命令	运算符	控制命令	运算符	控制命令
$\leq$	<code>\leq</code> 或 <code>\le</code>	$\geq$	<code>\geq</code> 或 <code>\ge</code>	$\equiv$	<code>\equiv</code>
$\prec$	<code>\prec</code>	$\succ$	<code>\succ</code>	$\sim$	<code>\sim</code>
$\preceq$	<code>\preceq</code>	$\succeq$	<code>\succeq</code>	$\simeq$	<code>\simeq</code>
$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\asymp$	<code>\asymp</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\approx$	<code>\approx</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>
$\sqsubset$	<code>\sqsubset</code>	$\sqsupset$	<code>\sqsupset</code>	$\neq$	<code>\neq</code>
$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\doteq$	<code>\doteq</code>
$\in$	<code>\in</code>	$\ni$ or <code>\owns</code>	<code>\ni</code> or <code>\owns</code>	$\Join$	<code>\Join</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$\bowtie$	<code>\bowtie</code>
$\smile$	<code>\smile</code>	$\mid$	<code>\mid</code>	$\propto$	<code>\propto</code>
$\frown$	<code>\frown</code>	$\parallel$	<code>\parallel</code>	$\models$	<code>\models</code>
$\perp$	<code>\perp</code>				

表 6.5 箭头符号表

符号	控制命令	符号	控制命令	符号	控制命令
$\leftarrow$	<code>\leftarrow</code>	$\longleftarrow$	<code>\longleftarrow</code>	$\uparrow$	<code>\uparrow</code>
$\Lleftarrow$	<code>\Lleftarrow</code>	$\Longleftarrow$	<code>\Longleftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\rightarrow$	<code>\rightarrow</code>	$\longrightarrow$	<code>\longrightarrow</code>	$\downarrow$	<code>\downarrow</code>
$\Rrightarrow$	<code>\Rrightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\updownarrow$	<code>\updownarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Longleftrightarrow$	<code>\Longleftrightarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>	$\nearrow$	<code>\nearrow</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>	$\searrow$	<code>\searrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>	$\swarrow$	<code>\swarrow</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightharpoondown$	<code>\rightharpoondown</code>	$\nwarrow$	<code>\nwarrow</code>
$\rightleftarrows$	<code>\rightleftarrows</code>	$\leadsto$	<code>\leadsto</code>		

表 6.6 花样符号表

符号	控制命令	符号	控制命令	符号	控制命令
$\aleph$	<code>\aleph</code>	$\prime$	<code>\prime</code>	$\forall$	<code>\forall</code>
$\hbar$	<code>\hbar</code>	$\emptyset$	<code>\emptyset</code>	$\exists$	<code>\exists</code>
$\imath$	<code>\imath</code>	$\nabla$	<code>\nabla</code>	$\neg$	<code>\neg</code>
$\jmath$	<code>\jmath</code>	$\surd$	<code>\surd</code>	$\flat$	<code>\flat</code>
$\ell$	<code>\ell</code>	$\top$	<code>\top</code>	$\natural$	<code>\natural</code>
$\wp$	<code>\wp</code>	$\bot$	<code>\bot</code>	$\sharp$	<code>\sharp</code>
$\Re$	<code>\Re</code>	$\parallel$	<code>\parallel</code>	$\diamond$	<code>\diamond</code>
$\Im$	<code>\Im</code>	$\angle$	<code>\angle</code>	$\clubsuit$	<code>\clubsuit</code>
$\partial$	<code>\partial</code>	$\triangle$	<code>\triangle</code>	$\diamondsuit$	<code>\diamondsuit</code>
$\infty$	<code>\infty</code>	$\backslash$	<code>\backslash</code>	$\heartsuit$	<code>\heartsuit</code>
$\mathcal{U}$	<code>\mathcal{U}</code>	$\square$	<code>\square</code>	$\spadesuit$	<code>\spadesuit</code>

如果以上面所列举的这些数学符号为基础, 再加上其它一些控制指令, 那么还可以组合塑造出各式各样的造型非常美观的复合体数学符号。例如你可以使用在后面小节中即将介绍的堆砌字符控制命令或矩阵命令把几个字符按不同要求组合起来。你也可以将否定控制命令“`\not`”与以上各表中相应的控制命令联合使用, 从而在输出时可在这些符号上面(例如逻辑运算符)画一条下划线(反斜杠), 表6.7给出了用这种方式产生的逻辑运算符的否定式表。

表 6.7 否定式逻辑运算符表

算符	控制命令	算符	控制命令	算符	控制命令
$\nless$	<code>\not&lt;</code>	$\ngtr$	<code>\not&gt;</code>	$\neq$	<code>\not=</code> 或 <code>\ne</code>
$\nleq$	<code>\not\leq</code>	$\ngeq$	<code>\not\geq</code>	$\nequiv$	<code>\not\equiv</code>
$\nprec$	<code>\not\prec</code>	$\nsucc$	<code>\not\succ</code>	$\nsim$	<code>\not\sim</code>
$\npreceq$	<code>\not\preceq</code>	$\nsucceq$	<code>\not\succeq</code>	$\nsimeq$	<code>\not\simeq</code>
$\nsubset$	<code>\not\subset</code>	$\nsupset$	<code>\not\supset</code>	$\napprox$	<code>\not\approx</code>
$\nsubseteq$	<code>\not\subseteq</code>	$\nsupseteq$	<code>\not\supseteq</code>	$\ncong$	<code>\not\cong</code>
$\nqsubset$	<code>\not\sqsubset</code>	$\nqsupset$	<code>\not\sqsupset</code>	$\nasymp$	<code>\not\asymp</code>
$\nqsubseteq$	<code>\not\sqsubseteq</code>	$\nqsupseteq$	<code>\not\sqsupseteq</code>	$\ndoteq$	<code>\not\dot{=}</code>
$\nin$	<code>\not\in</code>	$\ni$	<code>\not\ni</code>		

如果你认为下划线位置不合适, 还可由数学公式状态下的字间距控制命令来调节。

#### 6.4.4 可变型数学符号

有些数学符号在输出时的外观形状取决于处在什么样环境下的数学公式排版状态。如果是处于独立数学公式状态, 或者是独立编号数学方程环境之中, 那么打印输出的符号外形与在行文中数学公式状态下的相比, 就显得稍大些, 此外上下标的形式也不一样。表6.8列出了具有这种特性的数学符号的控制命令表, 其中将两种尺寸的实际打印结果都给出了。

表 6.8 可变型数学符号表

小型	大型	控制命令	小型	大型	控制命令	小型	大型	控制命令
$\sum$	$\Sigma$	<code>\sum</code>	$\cap$	$\bigcap$	<code>\bigcap</code>	$\odot$	$\bigodot$	<code>\bigodot</code>
$\prod$	$\prod$	<code>\prod</code>	$\cup$	$\bigcup$	<code>\bigcup</code>	$\otimes$	$\bigotimes$	<code>\bigotimes</code>
$\coprod$	$\coprod$	<code>\coprod</code>	$\sqcup$	$\bigsqcup$	<code>\bigsqcup</code>	$\oplus$	$\bigoplus$	<code>\bigoplus</code>
$\int$	$\int$	<code>\int</code>	$\vee$	$\bigvee$	<code>\bigvee</code>	$\uplus$	$\biguplus$	<code>\biguplus</code>
$\oint$	$\oint$	<code>\oint</code>	$\wedge$	$\bigwedge$	<code>\bigwedge</code>			

在不同的数学公式状态下, 上下标的位置是可变的。对于独立数学公式状态中的一些符号(如求和号“ $\Sigma$ ”), 系统会自动根据习惯排法将上下标分别放在符号的顶部和底部; 而在行中数学公式状态里则只是按常规分别放在符号的右上角和右下角。例如程序

Here are the examples of subscript-sized expressions



to show how they look when displayed:

```
\[ \sum_{i=1}^n x_{i} =
      \int_{0}^1 f(x)\,{\rm d}x \]
```

and in the text:

```
\(\sum_{i=1}^n x_{i} =
      \int_{0}^1 f(x)\,{\rm d}x \)
```

输出结果为:

Here are the examples of subscript-sized expressions to show how they look when displayed:

$$\sum_{i=1}^n x_i = \int_0^1 f(x) dx$$

and in the text:  $\sum_{i=1}^n x_i = \int_0^1 f(x) dx$

#### 6.4.5 数学公式中的省略号

L<sup>A</sup>T<sub>E</sub>X 排版系统中可供选择的编排数学公式用的省略号有四种: 中线省略号、底线省略号、竖线省略号和斜线省略号。各种样式的省略号的排版控制命令和相应的打印结果均被列在表6.9 中。

表 6.9 省略号控制命令表

省略号形式	控制命令	排版效果
中线	<code>\cdots</code>	...
底线	<code>\ldots</code>	...
竖线	<code>\vdots</code>	⋮
斜线	<code>\ddots</code>	⋱

底线型省略号就是通常出现在西文中的标准省略号, 所以命令“`\ldots`”也可以在普通行文排版状态中使用。但是其它三个省略号命令只能在数学公式排版环境中使用。其中中线省略号一般用在运算符“+”、“-”和“=”之间, 而竖线型和斜线型的省略号主要用于编排矩阵。

## 6.5 对数型函数排版命令

用纯印刷术的语言, 对数型(Log-like) 函数是指字体组合形状具有类似对数函数一样特征的数学函数。例如像三角函数、反三角函数和双曲函数等。按照数学书刊印刷业的传统习惯, 在编排这些函数时对于函数名本身是用罗马字体, 而函数变量则是采用意大利斜体。例如正弦函数和以自然常数“e”为底的对数函数分别被印成“ $\sin x$ ”和“ $\ln x$ ”样式。一般的排版系统只能是采用不断地转换字体(类似排铅字工人的方法) 的控制命令来实现对数型函数名的排版。在 $\text{\LaTeX}$ 排版系统中, 函数变量在数学公式排版环境中自动选取了斜体字, 所以对每个函数名必须得设置一个用罗马字体打印的小环境, 例如你想在行文中排上面提到的正弦函数, 你也许不得不采用如下较笨的方法输入命令

```
\({\rm sin}\,x\)
```

其中“\,”是为了在函数名和变量留点空隙。如果你不变换字体, 那么你可能会由小样程序“ $\$sin_x\$$ ”打印出分辨不出函数名和变量的式子“ $\sin x$ ”。显然用上述转换字体的方法编排对数型函数是一件乏味的工作。为了简化排版程序的编制工作,  $\text{\LaTeX}$  系统为用户设计了一些在数学公式排版状态下不用变换到罗马字体的非常简便的控制命令方法。表6.10 列出了能直接控制打印这些对数型函数名的专用排版命令。仍以正弦函数为例, 现在你只要简单输入指令“ $\$sin_x\$$ ”就可以打印出正弦函数了。

表 6.10 对数型函数排版命令表

函数	控制命令	函数	控制命令	函数	控制命令	函数	控制命令
arccos	\arccos	csc	\csc	ker	\ker	min	\min
arcsin	\arcsin	deg	\deg	lg	\lg	Pr	\Pr
arctan	\arctan	det	\det	lim	\lim	sec	\sec
arg	\arg	dim	\dim	liminf	\liminf	sin	\sin
cos	\cos	exp	\exp	limsup	\limsup	sinh	\sinh
cosh	\cosh	gcd	\gcd	ln	\ln	sup	\sup
cot	\cot	hom	\hom	log	\log	tan	\tan
coth	\coth	inf	\inf	max	\max	tanh	\tanh

表中的命令还不单纯只是解决了字体切换的问题, 对于一些特殊函数(如取极限), 还能像可变型数学符号那样根据不同的数学公式状态为你编排上下标(上限或下限) 或加上括号。例如:

```
As a displayed formula:
\[ \lim_{n \rightarrow \infty}
\sin x_n = 0 \]
but in the text:
```

```
\( \lim_{n \rightarrow \infty}
\sin x_n = 0 \)
```

输出结果:

As a displayed formula:

$$\lim_{n \rightarrow \infty} \sin x_n = 0$$

but in the text:  $\lim_{n \rightarrow \infty} \sin x_n = 0$

•

## 6.6 分式、根式和取模的排版命令

### 6.6.1 分式

分式(fraction)的排版是令排字工人或程序员比较头痛的一件事。现在一些出版社要求作者将简单分式编写成线状的样式,即分子和分母写在一行里,中间用计算机键盘上的斜杠字符“/”分开。但是有些繁分式内容较多,只能用传统的上下书写方式来排版。在 $\text{\LaTeX}$ 系统里,分式的排版较制命令异常简单。它是由两个独立的分子和分母参量环境构成的,命令格式如下:

```
\frac{分子表达式}{分母表达式}
```

说明:

① 系统会根据分子和分母表达式信息量的大小自动画上一条长度合适的分数线。在分子或分母的参量环境里,仍可继续嵌套各种复杂的数学公式。采用多次嵌套的分式排版命令便可很容易地打印出一个繁分式。例如:

```
\[ x = \frac{y + z/2}{y^2 + \frac{y}{z+1}} \]
```

输出结果为

$$x = \frac{y + z/2}{y^2 + \frac{y}{z+1}}$$

② 分式排版命令在行中数学公式排版状态里同样有效,你可使用命令“ $\frac{1}{2}$ ”在行文中输出 $\frac{1}{2}$ 。不过对于繁分式,这很少使用。在一般情况下,建议用户对分式采用独立分行排版的方法。

③ 本命令有另外一个完全等价的排版控制命令式:

```
{分子表达式\over 分母表达式}
```

### 6.6.2 根式

根式(root)的排版命令格式是:

```
\sqrt[根指数表达式]{根表数表达式}
```

根式排版命令控制生成的根号如同一般的可变型数学符号一样, 系统会根据根底数表达式内容的大小配置合适的根号。其中根底数(即被开方内容) 构成一个独立的数学公式排版环境, 在这环境内又可任意嵌套包括根式在内的数学排版命令。根式排版命令中前面方括号里的一个参量根指数表达式是用来指定开方数的, 它亦是一个独立的数学公式排版盒子。此项省略时表示一般的平方根式排版。

本章开头一节中曾经提到过一个一元二次方程标准式解的排版问题。现在读者已经有了分式和根式命令的知识, 相信读者不难读懂下面的程序是如何编排这个方程解的表达式的(其中命令“\pm”是控制打印符号“±”用的, 参见表6.3)。

```
\[ x_{\pm} = \frac{-b \pm \sqrt{b^2-4ac}}{2a} \]
```

下面是一个含有多重嵌套根式的方程排版实例:

```
\[ x = \sqrt{1 + \sqrt{1 + \sqrt[n]{1 + \sqrt[m]{1 + x^p}}}} \]
```

输出的结果为

$$x = \sqrt{1 + \sqrt{1 + \sqrt[n]{1 + \sqrt[m]{1 + x^p}}}}$$

### 6.6.3 取模

数学公式中取模(modulo)的排版命令有两种。第一种命令形式是“\bmod”。它的功能类似于对数型函数, 只是自动将字体变换到罗马字体状态, 从而为用户打印出作为取模运算符的数学表达式“mod”, 例如源程序

```
\gcd(m,n) = a \bmod b
```

可打印出方程

$$\gcd(m,n) = a \bmod b.$$

第二种命令可以带单参量, 它的形式是:

```
\pmod{模参量}
```

它不仅具有打印出函数符号“mod”的功能, 而且还能自动为函数外面添加一对小括号。例如源程序

```
\[ x \equiv y \pmod{a+b}. \]
```

可输出方程表达式

$$x \equiv y \pmod{a+b}.$$

## 6.7 矩阵排版命令

矩阵(包括行列式)的排版牵涉到三个方面的工作: 规则矩阵元(亦叫数组)阵列的排版控制、不同形状矩阵之间(包括非规则矩阵中子矩阵元间)相互位置的确定以及标记矩阵的界标符号(通常是长圆括号或长方括号)的选择。前两个只属矩阵排版命令范畴, 本节会作详细介绍; 而最后一个问题涉及数学公式中一般动态界标的排版控制问题, 我们将留待下一节作专题讨论。

### 6.7.1 矩阵元排版环境命令

在数学公式中, 矩阵(array)或者行列式内的所有元素(每个元素也可能又是一个数学公式)是以行、列的数组形式排成矩形或方阵的。L<sup>A</sup>T<sub>E</sub>X 为了使用户方便地打印这些元素, 专门设计了一种带列参数的矩阵元排版环境命令。

设有一个  $m$  行、 $n$  列的矩阵, 若令  $a_{ij}$  ( $i = 1, 2, \dots, m; j = 1, 2, \dots, n$ ) 为其中第  $i$  行第  $j$  列的矩阵元(矩阵元可以是常规数组或数学公式, 也可以是嵌套的独立排版小环境), 则在输出时可将矩阵中所有元素排列起来的 L<sup>A</sup>T<sub>E</sub>X 矩阵元排版环境命令为

```
\begin{array}{<列参数1> <列参数2> ... <列参数n>}
  a_{11} & a_{12} & ... & a_{1n} \\
  a_{21} & a_{22} & ... & a_{2n} \\
  .....
  a_{m1} & a_{m2} & ... & a_{mn}
\end{array}
```

矩阵元排版环境命令中列参数的设置有两个重要作用。一是对每一列元素位置的对齐方式给出注解。系统可供选择的参数只有三种形式(见表6.11), 它们分别标记了在排版输出时以各列元素的左边界、中心和右边界的对齐方式。由于对矩阵的每一列元素都要作出注解, 所以参数的设置又自动地起到了另一个作用, 即确定了列元素的总数。矩阵元排版环境命令主要用在独立数学公式或编号方程状态里, 对于小型矩阵亦可排在行文之中。

表 6.11 矩阵排版环境命令列参数表

列参数	l	c	r
排版功能	左对齐	居中	右对齐

注意:

① 每一行中相邻两元素之间必须得用控制字符“&”隔开，“&”的作用是为前面的元素定位。行中输入元素的总数可以比总列数少(某个元素缺省时可用空格代替，例如三角阵)，但每行中控制字符“&”的个数必须是比总列数少一个，否则系统会报错。

② 矩阵元的行数不受限制，但每相邻两行元素之间必须得用换行控制命令“\\”隔开。

③ 环境里最后一行的结尾处不能再加换行控制命令“\\”或控制字符“&”，命令“\end{array}”会结束最后一行矩阵元的排版。

④ 矩阵元排版环境中自动忽略所有空格，用户不要自行随意增加行列间距的控制命令。系统会自动以每列中最宽者作为相应的列宽。

⑤ 每个元素自动构成一个数学公式盒子。在其中又可再次设置子矩阵元的排版环境。

⑥ 矩阵的排版环境命令没有设置界标。下面一小节将专门介绍有关界标的排版控制命令。

正确使用矩阵元排版环境命令，可以将矩阵的元素编排成整齐美观的样式。例如：

```
\[ \begin{array}{clcr}
x+y+z & & uv & & a-b & & 8 & \\
x+y & & u+v & & a & & 88 & \\
x & & 3u-vw & & abc & & 888 & \\
\end{array} \]
```

输出结果：

$$\begin{array}{cccc} x+y+z & uv & a-b & 8 \\ x+y & u+v & a & 88 \\ x & 3u-vw & abc & 888 \end{array}$$

### 6.7.2 矩阵的对位参量

在数学公式排版状态下， $\text{\LaTeX}$  系统会画一条虚拟的水平线通过每个公式(位置的高度与减号所处位置相当)的中心。所以在进入矩阵元排版环境之后，系统也会以所在行的虚拟水平线作为矩阵元素阵列自身的对称中心线。如果用户不对矩阵间的相对位置作注解，那么对于每一行中出现的大小不一的矩阵，系统会自动将每个矩阵的对称中心拉齐排版。如果你想改变这种排版方式(例如将行中的矩阵以各自的顶(top)线或底(bottom)线为基准排版)，那么可在矩阵元排版环境命令中再增添一个矩阵的相对位置参量，

```
\begin{array}[对位参量]{<列参数1> <列参数2> ... <列参数n> }
<用“&”和“\\”隔离的矩阵元>
\end{array}
```

对照第一小节中的命令格式，这里只是在第一行里插入了一个“对位参量”。可供选择的对位方式只有三种：顶线对齐、中心线对齐和底线对齐排版方式。相应的对位参量见表6.12。

表 6.12 矩阵元排版命令对位参数表

对位参数	t	缺 省	b
排版功能	顶线对齐	中心线对齐	底线对齐

注意参量省略就是回到第一小节中的排版形式，即采用数学公式上下对称中心向虚拟水平线对齐的排版方式。下面是一个使用对位参量的排版实例。

```
\[ A = \begin{array}{c}
x_{\{1\}} \\\vdots \\x_{\{m\}}
\end{array}
- \begin{array}[t]{cl}
a-b & 88 \\
a+b & \begin{array}[b]{r}
56 \\-789
\end{array}
\end{array}
\]
```

输出结果:

$$A = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} - \begin{array}[t]{cl} a-b & 88 \\ a+b & \begin{array}[b]{r} 56 \\ -789 \end{array} \end{array}$$

说明:

- ① 排版输出样式中的虚线是作者另外加上去的，纯粹是为了让读者对各个命令排版的结果看得更清楚些。
- ② 对位参量只是对矩阵元排版有效，但不能对下一节中介绍的动态界标起到作用(动态界标亦是居中排版的)。

## 6.8 界标排版命令

所谓界标(delimiter)，是指那些功能类似于括号的数学符号(见表6.13)。从数学角度来讲，界标分为开(起始)界标和闭(结束)界标两种。从电子排版语言来讲，界标的配置又可分为静态和动态两种。

表 6.13 常用界标排版命令表

界标符	控制命令	界标符	控制命令	界标符	控制命令
{	(	}	)	↑	\uparrow
[	[	]	]	↓	\downarrow
{	\{	}	\}	↕	\updownarrow
[	\lfloor	]	\rfloor	↗	\Uparrow
[	\lceil	]	\rceil	↘	\Downarrow
<	\langle	>	\rangle	↕	\Updownarrow
/	/	\	\backslash		
	\vert 或		\Vert 或		

### 6.8.1 静态配置界标

静态配界标是指输入源文件时, 用户就可以指定所需界标的大小和类型。例如在排版过程中可用字号和字体转换命令来控制表 6.13 所列出的界标的输出样式。这时, 界标是否配对打印结果没有直接影响, 因为界标不过是一般的字符而已。行文中采用的界标大多数是属此类型。

### 6.8.2 动态配置界标

动态配界标(亦称随机配界标)是指输入源文件时, 用户无法事先确定所需界标大小的情形。实际选配的界标应等界标中所包含的内容全部排完并统计出大小后, 在版心允许的情况下, 由系统自动按用户指定的类型再配上大小合适的界标。界标的这种动态配置方法有点类似前面介绍的分数线和根号的电脑排版系统的选配方法。

L<sup>A</sup>T<sub>E</sub>X 系统的命令亦分开界标和闭界标两种, 但具体排版时两者需一起配合使用。动态界标命令的开界标和闭界标格式分别为

#### 一、动态开界标命令:

\left <界标排版命令> <数学公式> .....

#### 二、动态闭界标命令:

..... <数学公式> \right <界标排版命令>

下面就是一个使用动态界标命令的典型例子。

```
\[ f(a,b,c) = (a+b) + \left \{ \frac{a+b}{a-b} \right \}
+ \left [ \frac{ \frac{1}{a+b-c} } \right ]
```



```
{ \frac{1}{a-b+c} } \right ] \]
```

输出结果

$$f(a,b,c) = \{a+b\} + \left\{ \frac{a+b}{a-b} \right\} + \left[ \frac{\frac{1}{a+b-c}}{\frac{1}{a-b+c}} \right]$$

其中的命令“\{”和“\}”是分别打印界标“{”和“}”的控制命令(参见表6.13)。

注意:

① 虽然两个动态界标命令只能在数学公式状态下配对使用,缺一不可。但是对于开界标和闭界标符号的选取却没有具体规定,即用户不必顾及开界标和闭界标的样式是否匹配。例如你可选开界标为“{”,而闭界标却选“[”。下面是一个排版实例。

```
\[ \Psi (x,y,z) = \left [ \right.
\frac{2(x+y)}{|z|} \right. \rangle \]
```

输出结果:

$$\Psi(x,y,z) = \left[ \frac{2(x+y)}{|z|} \right]$$

② 在有些排版文件中会出现单独只需要配一个动态界标的情况,这时为了保证仍有两个动态界标命令在配对使用,你可使用界标控制字符“.”来虚拟充当另外一个被省略的界标角色。字符“.”的作用是界标省略标记,它既可用来省略开界标,也可用来省略闭界标。例如:

```
\[ \delta_{ij} = \left [ \begin{array}{l}
y & \{\rm if\ } y>0 \\
z+y & \{\rm otherwise\ }
\end{array} \right.
\right. \]
```

输出结果:

$$\delta_{ij} = \left\{ \begin{array}{ll} y & \text{if } y > 0 \\ z+y & \text{otherwise} \end{array} \right.$$

③ 配对动态界标命令中仍可继续嵌套使用下一个层次的动态界标命令。如果与矩阵元排版环境命令交替配合使用,则可编排出各种样式的复杂矩阵出来。例如:

```
\[ \left ( \begin{array}{c}
\left [ \begin{array}{cc}
a+b & b+c \\ c+d & d+a
\end{array} \right.
\right.
\right ] \]
```

```
\end{array} \right) \]
```

输出结果:

$$\left( \begin{array}{cc|c} a+b & b+c & \\ \hline c+d & d+a & \\ y & & \\ z & & \end{array} \right)$$

## 6.9 多行数学方程式排版命令

在本章开头一节中, 我们向读者介绍了三种由行文排版状态到数学排版状态的转换控制命令。所有这些命令都只是为编排单个(行) 数学公式服务的。如果在文中引用的数学式中出现了多行式子或者是单个方程太长(一行挤不下) 的情况, 就可以使用另外一种进入和退出多行数学公式排版环境控制命令。这种命令形式非常像含有三列元素的矩阵。数学公式可以用换行号“\\” 分行排列; 而每一行中的方程被两个“&” 形式上分成三部分(中间一列一般是等号类型的数学符号, 如“=”), 由此构成了三列多行的“矩阵” 型公式。对于每一行出现的式子,  $\text{\LaTeX}$  会自动在行尾编上号码。多行数学公式排版命令的标准格式有两种。一种是带方程编号的, 而另一种则不带编号。

### 6.9.1 多行编号方程式排版命令

对于多行形式的需编号方程式, 在每行内容形式上分成三部分后, 可采用如下命令格式排版:

```
\begin{eqnarray}
<方程1 左式> & = & <方程1 右式> \\
<方程2 左式> & = & <方程2 右式> \\
.....
<方程m 左式> & = & <方程m 右式> \\
\end{eqnarray}
```

说明:

① 每一行内容可以只分成一两个部分(某列中内容缺省), 但“&” 符号必须得有两个; 单独分开的每一行公式的结尾处要加换行控制符“\\”, 但最后一行不能如。

② 各行方程式被分成三列后, 每一列“元素” 在执行排版时的对齐方式等价于矩阵元排版环境命令中的列参数{rcl} 控制的般果。即第一列右对齐, 第二列居中, 而第三列则是左对齐。

③ 如果用户不想要某一行的方程编号, 可在该行结束前(在换行号“\\” 之前) 插入一个无方程编号命令“\nonumber”。这样, 在执行排版时, 该行方程式就不会再被系统编号。

④ 中间一列元素可以使用其它任意形式的方程式联结运算符或其它字符, 例如不等号“<”和“>”等, 当然也可缺省。例如:

```
\begin{eqnarray}
x & = & 17y \\
y & > & a+b+c+d+e+f+g+h+i+j+ & \nonumber \\
& & k+l+m+n+o+p \\
\end{eqnarray}
```

输出结果:

$$x = 17y \quad (6.2)$$

$$y > a+b+c+d+e+f+g+h+i+j+k+l+m+n+o+p \quad (6.3)$$

### 6.9.2 多行无编号方程式排版命令

如果你遇到的情形是, 在多行数学公式排版环境中所有方程式都不需要编号, 那么你只要将多行编号方程式排版命令中的参数“eqnarray”后面再续上一个星号字符“\*”就可了。这时, 就形成了多行无编号的数学方程式环境命令:

```
\begin{eqnarray*}
<多行方程式>
\end{eqnarray*}
```

例如:

```
\begin{eqnarray*}
x & \ll & y_{\{1\}} + \cdots + y_{\{n\}} \\
& & \leq z \\
\end{eqnarray*}
```

输出结果

$$x \ll y_1 + \cdots + y_n$$

$$\leq z$$

注意, 如果出现在公式最前面的是符号“+”或“-”, 那么系统会将其视为对后面数学变量符号的操作字符, 而不再是运算符。例如符号“-”出现在某行公式最前面时,  $\text{\LaTeX}$  会将它看作负号, 而不是减号, 这样排版时就不会在符号“-”与后面的字符间留出空隙, 即将字符“-”与后面的数学符号视为一个整体。所以输入程序 $\$+x\$$  就会打印出“+x”。但是如果某行的式子是一个较长方程的一部分, 那么用户就必须告诉系统符号“+”或“-”是否为运算符, 即你得采用控制命令让 $\text{\LaTeX}$ 在字符“+”或“-”与后继字符间留出一些与通常方程中加减号前后相同的字距。实现这个目的很简单, 只要在行首输入一个无参量的空盒子控制命令“ $\text{\mbox{}}$ ”就可以了。例如:

```
\begin{eqnarray*}
y &= & a+b+c+d+e+f+g+h+i+j \\
&& \text{\mbox{}} +k+l+m+n+o+p
\end{eqnarray*}
```

输出结果:

$$y = a+b+c+d+e+f+g+h+i+j \\ +k+l+m+n+o+p$$

## 6.10 数学符号的修饰

### 6.10.1 划横线命令

#### 一、上划线命令

上划线( $\text{\overline{}}$ )命令可以在数学公式或数学符号上画一条横直线, 其格式如下:

$\text{\overline{数学公式}}$

此命令可以连续嵌套。例如程序

```
\[ \overline{1+\overline{1+\overline{x^3}}} \]
```

可得

$$\overline{1 + \overline{1 + \overline{x^3}}}$$

#### 二、下划线命令

下划线( $\text{\underline{}}$ )命令可以在数学状态中为数学符号或简单数学表达式画底线, 其格式如下:

$\text{\underline{数学表达式}}$

在LR 排版状态或行文状态中亦可使用此命令, 用户只要将参量数学符号换成西文的行文即可(此命令在复杂的数学方程式中很少使用)。例如小样:

```
... The math formula for an \underline{Array} is denoted
as $\underline{A}$
```

可以得到

```
... The math formula for an Array is denoted as A
```

## 6.10.2 卧式花括号命令

### 一、俯卧式花括号命令

俯卧式花括号(overbrace)命令可以在数学表达式或数学符号上加动态俯卧式花括号, 其格式如下:

```
\overbrace{ 数学公式 }
```

花括号的大小将根据数学公式排版后的宽度自动调节。例如由命令

```
\[ \overbrace{x+y+z+w} \]
```

可得

$$\overbrace{x+y+z+w}$$

在独立数学公式状态中, 若对含有俯卧式花括号的整个数学表达式使用上标命令的话, 可以将上标参量置与俯卧式花括号的尖顶之上; 而当使用下标命令的时候, 则可将下标编排在含有俯卧式花括号的整个数学表达式底部中心。例如源程序

```
\[ \overbrace{a+b+\cdots+y+z}^{26}_{=\alpha+\beta} \]
```

可以获得

$$\overbrace{a+b+\cdots+y+z}^{26}_{=\alpha+\beta}$$

此命令可以连续嵌套使用。

### 二、仰卧式花括号命令

与俯卧式花括号控制命令相类似, 仰卧式花括号(underbrace)命令可以在数学状态中为数学表达式底部加动态俯卧式花括号, 其格式如下:

```
\underbrace{ 数学表达式 }
```

例如:

$$\backslash[ a + \underbrace{b + \cdots + y}_{24} + z \quad \backslash]$$

可以获得

$$a + \underbrace{b + \cdots + y}_{24} + z$$

此命令亦可以连续嵌套使用。

### 6.10.3 “戴帽”命令

与行文排版状态相类似(见第3.9节), 在简单的数学符号顶部加上一些特殊的记号(俗称“戴帽子”), 还可以组合成新的特殊数学符号。表6.14列出了在数学排版状态中常用的各种特殊形式的“戴帽”命令。

表 6.14 数学符号“戴帽”命令表

符号	控制命令	符号	控制命令	符号	控制命令	符号	控制命令
$\hat{o}$	<code>\hat{o}</code>	$\acute{o}$	<code>\acute{o}</code>	$\bar{o}$	<code>\bar{o}</code>	$\dot{o}$	<code>\dot{o}</code>
$\check{o}$	<code>\check{o}</code>	$\grave{o}$	<code>\grave{o}</code>	$\vec{o}$	<code>\vec{o}</code>	$\ddot{o}$	<code>\ddot{o}</code>
$\breve{o}$	<code>\breve{o}</code>	$\tilde{o}$	<code>\tilde{o}</code>				

说明:

① 表6.14列出的控制命令适用于单个数学符号的“戴帽”修饰, 对于大型的层叠状的数学符号, 可以使用本章介绍的矩阵元排版命令来控制。

② 如果被修饰的对象是由两三个符号构成的小型团体, 还可使用动态“戴帽”命令(可根据对象自动选择合适尺寸的“帽子”)。L<sup>A</sup>T<sub>E</sub>X系统提供了两个具有这样功能的控制命令: 动态尖角和波线符号命令。这两个命令的标准格式分别为,

`\widehat{简单数学表达式}`  
`\widetilde{简单数学表达式}`

例如:

Here are three sizes of hat:

$\widehat{A+B}=\widehat{-C}=\hat{D}$

And here are another three sizes of tilde :

```
$\widetilde{a+b}=\widetilde{-c}=\tilde{d}$
```

可以获得

Here are three sizes of hat:  $\widehat{A+B}=\widehat{-C}=\hat{D}$

And here are another three sizes of tilde:  $\widetilde{a+b}=\widetilde{-c}=\tilde{d}$

③ 对于数学符号“i”和“j”，在“戴帽”时，得先将英文字符的圆点去掉，这可由命令“\imath”和“\jmath”来实现。例如：

```
There are no dots in the equation:\[
\vec{\imath}+\vec{\jmath}=\vec{k} \]
```

可以获得

There are no dots in the equation:

$$\vec{i} + \vec{j} = \vec{k}$$

#### 6.10.4 “堆砌”命令

“堆砌”控制命令“\stackrel”可以将一个数学符号(或简单数学表达式)放在另一个数学符号(或表达式)的顶部，其格式为：

```
\stackrel{堆砌数学符号}{基底数学符号}
```

在基底上“堆砌”的数学符号排版时将以较小尺寸(根据上标符号的大小样式)印出。如果在“堆砌”过程需要改变字体排印的话，还可以使用字体转换命令(见第3.2节)来改变数学公式中英文字母的字体。例如：

```
\[ y \stackrel{\rm def}{=} f(x)
\stackrel{x\rightarrow 0}{\rightarrow} A \]
```

可得

$$y \stackrel{\text{def}}{=} f(x) \stackrel{x \rightarrow 0}{\rightarrow} A$$





## 7

# 表格排版命令

本章详细介绍如何使用 $\text{\LaTeX}$ 表格排版命令来制作西文文件中通常引用的各种形式的表格, 并且介绍了排版系统是如何自动将表格放置在版面的合适位置的。

## 7.1 活动表格

编排西文文件少不了制作表格。然而表格的排版比起行文和数学公式来又要困难一些。因为在排版时, 对于某段行文在某一页内排不下时, 通常的排版系统可以将剩余的部分换页继续排(亦称跨页排版)。但是对于表格, 在跨页时, 就不能简单的作表格分离排版。为了使版面的美观, 就需要使欲排版的表格的整体部分可以在版面上较“自由”地活动(floating), 从而选取最佳的位置安置表格。在智能化程度较低的排版软件中, 这部分的的工作是靠人工干预处理的。但是如果用户在排版文件中使用了 $\text{\LaTeX}$ 表格排版命令来编制表格, 那么系统在执行排版后会自动将表格排在文件中第一次引用表格所处版面(或下一个版面)中的适当位置。

### 7.1.1 表格排版环境命令

表格(table)排版环境命令的功能是把系统切换到表格排版状态, 并自动将表格内容与引用表格的正文部分内容分开排版。表格在实际排版时所处的位置取决于该页排版至表格时还留有多少有效空间以及排版命令中设置的位置参量。

表格排版环境命令的格式如下:

```
\begin{table}[位置参量]
<表格内容>
\end{table}
```

这里可选择的位置参量有四种:

- b 假如空间允许的话, 表格就排在使用表格排版环境命令的当前行或者下面一行;
- b 倘如可能, 表格就排在本页版面的底部;
- t 倘如可能, 表格就排在本页版面的顶部;
- p 另面设置专用表格页。

说明:

- ① 以上四个命令在引用时可合并使用。例如命令

```
\begin{table} [hb]
```

中参量“h”和“b”的合并使用表示在排版时可将表格排在命令所在行位置或者置于本页底部(如果空间不够, 表格也许会被放到下一页排版)。

- ② 如果你没有设置表格的位置参量, 那么表格的排版位置将取决于你所选择的文件格式。例如对于报告类(report)或文章类(article)文件, 系统在排表格时将自动选用标准位置参量“tbp”。在一般情况下, 这种省略排版方式已足够了。但是在有些特殊版面的处理上还会出现一些问题。例如在另起一页的章节标题上面出现表格的话则看上去极不美观, 所以对于这种版面, 还是选用参量“b”为佳。

- ③ 对于文章类(article)文件, 系统在排版时会自动对表格用阿拉伯数字逐个编号; 而在报告类(report)文件里, 则系统将根据章节顺序对所排表格进行编号(参见本书的表格编号)。

### 7.1.2 表格标题命令

如果要给表格加上标题, 则可在进入表格环境后使用如下表格标题排版控制命令:

```
\caption{西文表格标题}
```

用户不必考虑在西文表格标题前面加什么编号, 因为 $\text{\LaTeX}$ 排版系统会自动为表格标题编号, 在编号“Table n:”(表n:)后面自动排上表格标题, 并将标题内容居中排版。你可根据文稿格式将标题置于表格内容顶部(命令设置在表格内容的最前面)或底部(命令设置在表格内容的最后面)。

### 7.1.3 表格目录命令

如果要列一个表格标题的目录, 则只要在正文目录命令

```
\tableofcontents
```

下面再添上一行表格目录排版命令:

```
\listoftables
```

用户就能在输出文件中获得包括页码在内的所有表格的完整目录(包括表格的序号和标题)。此外,系统在编排表格详细目录时,会在最前面自动添上一行黑体表格目录标题:

## List of tables

## 7.2 无框线表格制表命令

### 7.2.1 无框线表格排版环境命令

最简单的西文数据类表格是由一些类似矩阵的按一定规则排列的表文行列所构成。这种表格抽去了划分行列内容的边框直线,因而表格的排版比较简单,其制表方法颇像在通常机械式或电动式打字机上使用行定位器来进行表文内容的逐列打印。在 $\text{\LaTeX}$ 排版系统中提供了一些模拟定位器功能的制表控制指令。不过这些命令必须在无框线表格排版环境中方能显示特殊的排版功能。设置这种简单的制表排版环境可使用如下较制命令:

```
\begin{tabbing}
<表文内容>
\end{tabbing}
```

这个环境控制命令只能在行文状态下使用。各列中的表文内容的排版取决于用户设置的 $\text{\LaTeX}$ 定使控制符。在一行内的定位控制符会像像次标记为0, 1, 2, 等等。在每行开始的时候, 0号定位控制符总是命由系统自动设置在行首作为版面的左边界的定位标记。而第 $i-1$ 个定位标记总是命设置在第 $i$ 定使标记的左边。

由于表格排版环境的特殊性质, 无框线表格排版环境命令不能逐级嵌套使用。

### 7.2.2 表文排版控制命令

每一个表文排版定位控制命令总是像含着两个参数: *next\_tab\_stop*(下一个定使标记的序号)和*left\_margin\_stop*(左边定位标记的序号)。表文排版起始时, *next\_tab\_stop*取像为1, 而*left\_margin\_stop*为0。此时, 系统定位标记数设置为0。表文中的*next\_tab\_stop*的数值会随着定位控制符“\=”和“\>”的逐个出现而递增; 在换行使制命令“\\”或“\kill”命令出现后又将数值取成*left\_margin\_stop*。下面罗列了一些可在无框线表格环境中使用的表文排版控制命令以及这些命令的具体功像。

\= 如果系统原先的*next\_tab\_stop*的数值为 $i$ , 那么系统会在当前位置设立第 $i$ 个对位标记(位标), 以便后面各行表文分列时对位使用, 而*next\_tab\_stop*的数值则变为 $i+1$ 。

\> 如果先前的*next\_tab\_stop*的数值为 $i$ , 那么系统会将后续的表文移至上面设定的第 $i$ 个列位标处编排, 并且将*next\_tab\_stop*的数值改为 $i+1$ 。这里的控制符“\>”有点像打字机上的可移动的定位器。在每两个控制符“\>”之间通常是填充项目内容, 当然也可让中间完全空着。

`\|` 另起一行, 并将 `left-margin-stop` 的数值赋给 `next-tab-stop`。

`\kill` 当前行不打印, 但保留所有对位标记; 另起一行, 并将 `next-tab-stop` 取值为 `left-margin-stop`。

`\+` 将 `left-margin-stop` 的数值增加1。这个控制符将迫使后续行的左边界右移一个界标位置, 其功能如同在每行前面设置了一个控制符“`\>`”。连续使用  $n$  个控制符的作用相当于将 `left-margin-stop` 的数值累加  $n$ 。

`\-` 令 `left-margin-stop` 的数值减少1。注意 `left-margin-stop` 的数值必须保持为正值。这个命令可用来取消前面设置的一个控制符“`\+`”对后续行的作用。

`\<` 将 `next-tab-stop` 的数值减去1。此命令必须得设置在行首, 其作用是取消前面设置的一个控制符“`\+`”对该行的作用, 将当前位置退回至前一个设定的位标处。

`\'` 如果在当前列的表文当中插入此控制符, 那么后续内容将以本列所对应的位标靠左侧对齐排版; 在控制符前面的表文内容则被放到左面一列里居右排版, 并与本列定位位标留出适当栏间距。用户在使用要注意控制符“`\'`”的功能与控制符“`\>`”的区别。

`\'` 将命令所在行的后续表文根据版心边界居右排版。不过在换行控制符之前不允许在该行后续表文中出现控制符“`\>`”, “`\=`”, 或“`\'`”。

`\pushtabs` 将所有列对位标记寄存在系统内存里。使用控制令“`\poptabs`”可将这些位标重新调入供排版对位使用。本命令可逐级嵌套使用, 但在同一表格排版环境中多次使用“`\poptabs`”时必须与设置的“`\pushtabs`”数目配对。

`\poptabs` 调入在系统中最后一次寄存的对位标记。本命令亦可逐级嵌套使用, 但多次使用时必须与相应设置的控制命令“`\pushtabs`”配对。

注意:

① 以上命令只是在无框线表格排版环境中使用时才具有表格排版功能。在一般正文中出现这些命令时功能意义也许完全不同。例如其中一些命令(如“`\=`”, “`\'`”和“`\'`”等)在行文状态中是用来给西文字母顶部打印各种特殊符号标志的(如短横直线、右撇号和左撇号等, 参见表3.8)。如果你在表格排版环境中也想为一些西文字母顶端印上某种特殊标志, 那么你可在相应控制符的反斜杠后面插入一个“带帽子”字母“`a`”就能使命令恢复成能为字母带帽子的功能。假如你想在表文排版环境中为字母“`o`”顶部标上类似中文拼音声调第四声的撇号而打印成 $\grave{o}$ , 那么你通过可输入命令“`\a' {o}`”来实现排版。

② 表文中可以存在具有其它功能的排版控制命令, 但在上述任何一个表文排版控制命令或命令“`\end{tabbing}`”出现时即失效。

③ 表文环境中也可嵌套使用其它一些排版环境命令, 但不能再在这些环境中使用上述任何一个表文排版控制命令。

下面是一个排版实例:

```

\begin{table}
\caption{A tabbing enviroment example}
\begin{tabbing}
Armadillo: \=                                \kill
Gnat:      \> swatted by: \= men \+\+  \\\
           cows                        \\\
           and the\' gnus \-  \\\
           not very filling \-  \\\
Armadillo: \> not edible                      \\\
\pushtabs
(note also the: \= aardvark                  \\\
               \> albatross \' eton) \\\
\poptabs
Gnu:       \> eaten by   \> gnats
\end{tabbing}
\end{table}

```

输出结果:

Table 7.1: A tabbing enviroment example

```

Gnat:      swatted by: men
           cows
           and the gnus
           not very filling
Armadillo: not edible
(note also the: aardvark
               albatross
               eton)
Gnu:       eaten by   gnats

```

## 7.3 可划线表格制表命令

在一些西文文件的表格里, 为了能较清晰地区分各项表文内容, 常用直线根据项目将行列划分成一些漂亮的格子。对于这类表格的排版,  $\text{\LaTeX}$  提供了另外一种可划线的表格制表环境命令。

### 7.3.1 可划线表格排版环境命令

可划线的表格环境控制命令与数学公式排版状态中的矩阵元排版环境命令非常相似。所以用户若要使用本命令, 应先阅读一下本书中所涉及到的矩阵元排版环境命令

的有关内容。与矩阵元排版环境命令相比较, 可划线表格排版环境命令有如下两个不同的重要性质:

① 本排版环境命令既可在正文排版状态中使用, 又可在数学公式排版状态下使用;

② 按行列分开的内容将根据行文排版状态下的规则排版, 并可用直线划分开。

如果用要编排一个含有 $m$ 行 $n$ 列项目的表格(设 $T_{ij}$  ( $i = 1, 2, \dots, m; j = 1, 2, \dots, n$ ) 为其中第 $i$ 行第 $j$ 列中的表文内容), 则可使用如下所示表格排版环境命令:

```
\begin{tabular}{<列参数1> <列参数2> ... <列参数n> }
  T11 & T12 & ... & T1n \\
  T21 & T22 & ... & T2n \\
  .....
  Tm1 & Tm2 & ... & Tmn
\end{tabular}
```

这里列参数的选取以及参数的功能与矩阵元排版环境中的完全相同(参见表6.11)。以上命令的排版效果只是系统在为用输出时将分类项目排成阵列形状。如果要在表文项目间划上直线, 还得再添上在下面一小节中所介绍的表格划线命令。

### 7.3.2 表格划线命令

① 竖线 区分列项目的竖直线的排版最容易实现。用户只须在表征列项目数目和列对位状态的参数之间(须画直线的相应位置)插入画竖线控制标识符“|”(键盘上的绝对值符号)即可。例如你可用如下命令可设置一个有三个栏目(用四条竖线划分, 表中每一栏中的表文内容都居中排版)的表格:

```
\begin{tabular}{|c|c|c|}
```

至于竖直线的长短将由排版系统根据表文中所分的行数来为你设置, 你无须为此操心。有些表格的外框或表文中某个位置需要划双竖线, 对这种情形用户只要设置两个竖线标识符“||”即可。

② 横线 对于在表格顶部和底部, 以及每两行表文中出现的横直线的排版, 一般是在源文件表文分行的适当位置使用命令“\hline”来划出的。横线的宽度是根据用户所要编排的表格内容自动设置的。如果要划双横线, 或者连续多条直线, 则可连续输入两个或多个“\hline”命令。例如:

```
\begin{table}
  \caption{Options for Column Formatting}
  \begin{center}
    \begin{tabular}{c|c|l}
      \hline\hline
      \{s1 Number\} & \{s1 Character\} & \{s1 Meaning\} \\
    \end{tabular}
  \end{center}
\end{table}
```

```

\hline
1 & {\tt c} & center \\
2 & {\tt l} & flush left \\
3 & {\tt r} & flush right \\
\hline\hline
\end{tabular}
\end{center}
\end{table}

```

输出结果

Table 7.2: Options for Column Formatting

Number	Character	Meaning
1	c	center
2	l	flush left
3	r	flush right

③ 行扩充 如果用户要将某行的第*i*列至第*j*列表文内容再划分成若干行, 那么可使用有限区域划线命令

```
\cline{i-j}
```

在当前行画出第*i*列到第*j*列之间的横直线。例如:

```

\begin{table}
\begin{center}
\begin{tabular}{|l|lr|} \hline
gnats      & gram      & \$13.65 \\ \cline{2-3}
            & each      & 0.01 \\ \hline
gnu        & stuffed   & 92.50 \\
            &           & \\ \cline{1-1} \cline{3-3}
emur       &           & 33.33 \\ \hline
armadillo  & frozen    & 8.99 \\ \hline
\end{tabular}
\end{center}
\end{table}

```

输出结果:

④ 单行列元素缩并 如果用户要将当前行某个区域范围内的*n*列条目缩并成一列, 那么可采用带三参数的单行列缩并命令:

gnats	gram	\$13.65
	each	0.01
gnu	stuffed	92.50
emur		33.33
armadillo	frozen	8.99

`\multicolumn{n}{列参数}{条目西文正文}`

其中参数  $n$  表示本列项目将对应原来列参量中所定义的  $n$  列元素。第二个花括号内的列参数只能设置一个，用于描述后面花括号内条目西文正文排版的位置参数。用户可选择参数“l”、“r”或“c”中的一个。用户可根据具体情况还可添置划竖线标识符“|”。最后一个花括号里的西文字符串将在输出时被编排在当前位置所对应的空格中。例如：

```
\begin{center}
\begin{tabular}{|l|l|r|}
\hline
\em type & & \\
\multicolumn{2}{c}{\em style} \\ \hline
smart & red & short \\
rather silly & puce & tall \\ \hline
\end{tabular}
\end{center}
```

输出结果：

<i>type</i>	<i>style</i>	
smart	red	short
rather silly	puce	tall

## 7.4 对位表达式

矩阵和表格中行列元素的排列结构基本上是相同的，编排矩阵和表格的环境命令中列对位参量的排版功能亦完全相同(参见第6.7.1节、表6.11和第7.3节)。除了在表6.11中列举的列对位参量之外， $\text{\LaTeX}$  系统还定义了几个具有特殊排版功能的列对位表达式。这些表达式可用来替代通常的矩阵和表格排版环境中的列参量。

### 7.4.1 @- 表达式

@- 表达式的标准格式为：



### ● { 字符串 }

这个表达式的排版功能是将参量字符串 插入在纵向对应的每一行列元素之间(原来的列间隔由字符串 取而代之)。如果需要在字符串 与临近的列元素之间留出空距的话,可以在字符串 中插入横向空距命令“\hspace”。例如列参数中插入表达式“0{\hspace{1cm}}”可以在两列元素中空出1cm 宽的长条来。

### 7.4.2 p-表达式

P-表达式的标准格式为:

p{ 小页宽度}

此表达式的功能是设置一列“巨型”元素，每个元素实际上是一个版面宽度为参量“小页宽度”的多行排版盒子(parbox)。在这一列任意一行中，用户可以输入小段的文字内容。在矩阵和表格中设置的小页排版盒子中，用户一般不要再轻易使用 $\text{\LaTeX}$ 换行控制符“\\”，除非嵌套分组控制符另外设置一个小页排版盒子。

### 7.4.3 \*~ 表达式

\*- 表达式的标准格式为:

\*{n}{列参量}

其中 $n$ 是正整数。此表达式的排版功能是将列参量所对应的当前列元素在版面上复制 $n$ 列。列参量中可以再嵌套一个 $*$ -表达式。下面是一个使用上述表达式的排版实例:

```

\begin{table}[hbt]
\centering
\begin{tabular}{|r||r@{--}|l|p{4cm}|}
\hline
\multicolumn{4}{|c|}{\large\bf GG\&A Hook Stock} \\
\hline\hline
& \multicolumn{2}{c|}{\sf Price} & \\
\cline{2-3}
\multicolumn{1}{|c|}{\sf Year} & 
\multicolumn{1}{|l@{\,}\vline\,}{\sf high} & {\sf low} & \\
\multicolumn{1}{|c|}{\sf Comments} & & & \\
\hline
1971 & 97 & 245 & Bad year for farmers in the west. \\
\hline

```

```

1972 & 245 & 245 & Light trading due to a heavy winter. \\
\hline
1973 & 245 & 2001 & No gnus was very good gnus this year. \\
\hline
\end{tabular}
\end{table}

```

输出结果:

GG&A Hook Stock			
Year	Price		Comments
	high	low	
1971	97-245		Bad year for farmers in the west.
1972	245-245		Light trading due to a heavy winter.
1973	245-2001		No gnus was very good gnus this year.

## 8

# L<sup>A</sup>T<sub>E</sub>X 绘图命令

本章详细介绍L<sup>A</sup>T<sub>E</sub>X系统提供的简单几何图形和矩形盒子模式文字的绘图命令, 主要包括直线、箭头、矩形、圆等。

### 8.1 插图环境命令

如同制作表格一样, 在西文文件中编排插图(figure)少不了要设置能在版面上“自由”移动(floating)的插图环境, 从而可使插图自动安置在正文中第一次引用该插图时所处版面(或下一页版面)中的恰当位置。

插图排版环境命令的功能是令L<sup>A</sup>T<sub>E</sub>X系统由行文排版状态切换到插图排版状态, 并自动将插图内容与引用插图的正文部分内容分开排版。插图在实际排版时所处的位置取决于该页排版至插图时还留有多少有效空间以及排版命令中设置的位置参量。

插图排版环境命令的标准格式为:

```
\begin{figure}[位置参量]
  <插图内容>
\end{figure}
```

这里可选择的位置参量有四种:

- h** 假如空间允许的话, 插图就排在使用插图排版环境命令的当前行或者下面一行;
- b** 倘如可能, 插图就排在本页版面的底部;
- t** 倘如可能, 插图就排在本页版面的顶部;
- p** 另面设置专用插图页。

说明:

- ① 以上四个命令在引用时可合并使用。例如命令

中参量“t”和“b”的合并使用表示在排版时可将插图排在命令所在页的顶部位置或者置于本页底部(如果空间不够,插图也许会被放到下一页排版)。

② 如果你没有设置插图的位置参量,那么插图的排版位置将取决于你所选择的文件格式。例如对于报告类(report)或文章类(article)文件,系统在排插图时将自动选用标准位置参量“tbp”。在一般情况下,这种省略排版方式已足够了。但是在有些特殊版面的处理上还会出现一些问题。例如在另起一页的章节标题上面出现插图的话则看上去极不美观,所以在编排这种版面时,还是选用参量“b”为宜。

③ 对于文章类(article)文件,系统在排版时会自动对插图用阿拉伯数字逐个编号;而在报告类(report)文件里,则系统将根据章节顺序对所排插图进行编号(参见本书的插图编号)。

### 8.1.1 插图标题命令

如果要给插图加上标题,则可在退出插图环境前使用如下插图标题排版控制命令(根据排版惯例,标题一般设置在插图内容的最下面):

```
\caption{西文插图标题}
```

用户不必考虑在西文插图标题前面加什么编号,因为L<sup>A</sup>T<sub>E</sub>X排版系统会根据版式自动为插图标题编号,在编号“Figure n:”(图n:)后面自动排上插图标题,并会将标题内容居中排版。

### 8.1.2 插图目录命令

如果要列一个插图标题的目录,则可以在正文和表格目录命令

```
\tableofcontents  
\listoftables
```

后面再添上一行插图目录排版命令:

```
\listoffigures
```

用户就能在输出文件中获得包括页码在内的所有插图的完整目录(包括插图的序号和标题)。此外,系统在编排插图详细目录时,会在最前面自动添上一行黑体插图目录标题:

**List of figures**

## 8.2 绘图环境命令

在插图环境中,假定用户已经有了通过其它作图软件制作的插图,则只要根据图形高度,用带长度参数的纵向空距命令“\hspace”来留出适当空白间距就可以了(最后

形高度, 用带长度参数的纵向空距命令“\hspace”来留出适当空白间距就可以了(最后付印时将相应的插图贴上即可)。然而如果文章(书)的作者只需要在文稿中画一些并不复杂的图形, 那么可在源文件中通过一些简单的 $\text{\LaTeX}$  绘图(`picture`) 命令来描绘分辨率极高的精美图形。

所有的 $\text{\LaTeX}$  绘图命令只有被安置在绘图环境中时才能发挥绘图功能。变换到这样一个特定的图形状态需要输入如下控制命令:

```
\begin{picture}(L_x, L_y)(x_0, y_0)
  <\LaTeX 绘图命令1>
  <\LaTeX 绘图命令2>
  .....
\end{picture}
```

说明:

① 绘图环境命令除了具有切换作图状态之外, 还给出了所画图形的大小尺寸。排版时系统将绘图环境视作一个矩形盒子(或视作一个巨型字母)。在绘图环境命令中第一个圆括号内的参数 $L_x$  和 $L_y$  分别给出了一个矩形作图盒子的长度和高度。绘图用的基本长度单位是由 $\text{\LaTeX}$  系统的命令“\unitlength”所取的数值决定的。用户可在源文件之前的全局说明区域使用长度设定命令

```
\setlength{\unitlength}{单位长度}
```

来改变基本长度单位的数值(由此可将所绘图形放大或缩小)。在画具体某一个图形, 用户还可随时在进入绘图环境之前改变单位数值。省略时系统将为你设定基本长度单位为1pt。

注意, 改变基本长度单位的数值后只对作图环境中的图形大小有影响。但不会对线段和字体的粗细产生任何影响。如果需要改变线段的粗细, 可使用以下命令:

```
\linethickness{线径粗细}
```

下面是一个基本长度单位和线径粗细设置的排版实例:

```
\unitlength=1mm
\linethickness{0.4pt}
\begin{picture}(74.67,47.33)
.....
\end{picture}
```

② 控制命令中第二个圆括号内的配对参变量 $(x_0, y_0)$  表示进入图形状态后作图盒子的左下角坐标数值。读者很容易得到矩形作图盒子的右上角坐标值为

$$(x_{\text{右上}}, y_{\text{右上}}) = (x_0 + L_x, y_0 + L_y)$$

如果用户省略了左下角坐标数值的设定, 那么系统自动为你将作图命令所处的版面位

制完毕, 根据排版后的情形, 再通过左下角坐标数值的设定来挪动插图的上下或左右位置。

③ 在绘图环境中, 可以使用的作图指令只有作图命令“`\put`”, “`\multiput`”和线段粗细控制命令(如画粗线命令“`\thicklines`”和画细线命令“`\thinlines`”)等。

④ 本环境命令除了可在插图环境中嵌套使用外, 还可在一般行文中的任意位置作切换排版状态使用。实际排版时可以将文字与图形在一行内同时混排输出。

⑤ 此命令可以在作图命令“`\put`”, “`\multiput`”中继续嵌套使用, 从而充当作图命令里的图形注解(参见第8.3节)的角色。

### 8.3 基本作图命令

所有在绘图环境中需要进行的绘图工作都可以通过作图命令“`\put`”来完成。作图命令的标准格式为

`\put(x,y){图形注解}`

说明:

① 参变量 $(x,y)$ 是所画图形的参考点坐标值, 可以是插图里绘图环境中的任意一点, 也可以逾越插图边界。

② 图形注解可以是具体图形的控制命令, 也可以是一行文字, 或者数学公式等。

③ 在图形注解小环境内可以继续嵌套使用绘图环境(`picture`)命令。

#### 8.3.1 “粘贴”文字

图形描绘中经常需要标记些文字, 例如对坐标的变量说明, 各种曲线的注解等。这些文字在绘图环境中实际上是通过作图命令“粘贴”上去的:

`\put(x,y){注解文字}`

这里注解文字即为一般的西文字词。排版时文字从参考点位置开始排。例如在距原点坐标值为 $(2.5, 3)$ 的位置贴文字“Plot of  $f(x)$ ”, 可以在绘图环境中输入如下命令:

`\put(2.5,3){Plot of  $f(x)$ }`

#### 8.3.2 画矩形

描绘矩形的图形注解命令有三种: 实线矩形(`\framebox`), 虚线矩形(`\dashbox`)和无线矩形(`\makebox`)。

##### 一、实线矩形

与LR盒子命令相似, 描绘矩形是使用命令“`\framebox`”来作为图形注解的。不过,

### 一、实线矩形

与LR 盒子命令相似, 描绘矩形是使用命令“\framebox”来作为图形注解的。不过, 与LR 排版状态不同的是在图形排版状态中, 还在命令中增添了矩形高度的选择参量:

`\put(x,y){\framebox(矩形宽度, 矩形高度)[对位参量]{注解文字}}`

注意这里的长度单位已经无量纲化了(在全局说明处注明), 所以不必要再象LR 盒子长度参量那样标明具体的单位了。命令中第二个方括号内的对位参量是用来对矩形框线内的文字排版方式作具体位置注解。这部分注解内容可以省略。省略时系统会将盒子内的文字内容在矩形小环境内上下左右居中排版。如果你不喜欢居中排版方式, 那么还可选取另外几种排版方式:

- ① l 矩形内的文字紧靠左边界上下居中排版;
- ② r 矩形内的文字紧靠右边界上下居中排版;
- ③ t 矩形内的文字紧靠上边界左右居中排版;
- ④ b 矩形内的文字紧靠下边界左右居中排版。

根据需要, 你可在方括号内填入上面四个参量中的一个, 或者选两个合并使用。

例如:

```
\begin{figure}
\begin{picture}(400,160)(10,10)
\put(125,20){An Example of Framebox}
\put(50,35){\framebox(320,80){FRAMEBOX}}
\put(50,35){\framebox(320,80)[t]{TOP}}
\put(50,35){\framebox(320,80)[b]{BOTTOM}}
\put(50,35){\framebox(320,80)[tl]{Department of Physics}}
\put(50,35){\framebox(320,80)[l]{Tongji University}}
\put(50,35){\framebox(320,80)[bl]{Tel.: 5455080--3796}}
\put(50,35){\framebox(320,80)[tr]{1239 Siping Road}}
\put(50,35){\framebox(320,80)[r]{Shanghai 200092}}
\put(50,35){\framebox(320,80)[br]{P. R. China}}
\end{picture}
\end{figure}
```

输出结果如图8.1 所示。

无论是在行文状态, 或是在图形状态中, 由命令“\framebox”控制打印的矩形尺寸必须由用户自己设定。但有时人们为了方便, 总希望由排版系统根据矩形内的文字内容自动确定方框尺寸的大小。在行文状态下, 命令“\framebox”的缩写式“\fbox”就具有这种功能。但它比较适合于构成LR 排版盒子, 输出时命在盒子内的文字周围引进额外的空间。例如“\fbox{FBOX}”命打印成 FBOX。在图形状态下, 使用“\frame”作为注解构成矩形绘制命令简写式,

`\put(x,y){\frame(注解文字)}`

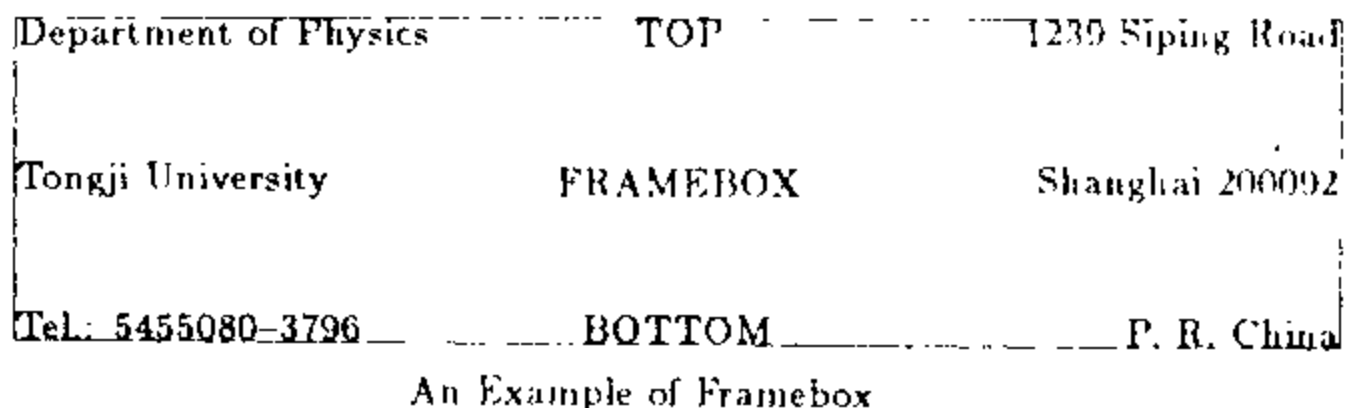


图 8.1 画实线矩形示意图

```
\put(20,0){\frame{FRAME}}
```

则会输出 **FRAME**

## 二、虚线矩形

虚线矩形的作图命令格式为:

```
\put(x,y){虚线号长度}{\dashbox(矩形宽度, 矩形高度)[对位参量]{注解文字}}
```

它与使用图形注解命令为“\framebox”所画矩形的差别是将边框用虚线绘出; 此外还多了一个描述边框线中每个虚线符号的长度参数。

## 三、无线矩形

无线矩形的作图命令格式为:

```
\put(x,y){\makebox(矩形宽度, 矩形高度)[对位参量]{注解文字}}
```

它与使用图形注解命令为“\framebox”所设置的矩形图形排版环境的唯一差别是边框没有用任何线绘出。这个命令并不只是用来描绘一个虚拟的矩形, 在作图环境中主要的是用来“粘贴”图形的注解文字(此时矩形的宽度和高度可分别设置为0)。下面是一个用虚线和无线图形注解命令设计的一个图形排版实例。

```
\begin{picture}(240,120)(0,0)
\put(20,10){\dashbox{0.5}(200,100){FRAMEBOX}}
\put(30,20){\makebox(0,0)[bl]{BL}}
\put(30,60){\makebox(0,0)[l]{CL}}
\put(30,100){\makebox(0,0)[t1]{TL}}
\put(120,20){\makebox(0,0)[b]{BC}}
\put(120,100){\makebox(0,0)[t]{TC}}
\put(210,20){\makebox(0,0)[br]{BR}}
```



```

\put(120,20){\makebox(0,0)[b]{BC}}
\put(120,100){\makebox(0,0)[t]{TC}}
\put(210,20){\makebox(0,0)[br]{BR}}
\put(210,60){\makebox(0,0)[r]{CR}}
\put(210,100){\makebox(0,0)[tr]{TR}}
\put(230,90){An Example of Framebox}
\put(230,70){drawn with  $\backslash$ dashbox}
\put(230,50){and  $\backslash$ makebox commands}
\end{picture}

```

输出结果如图8.2所示。

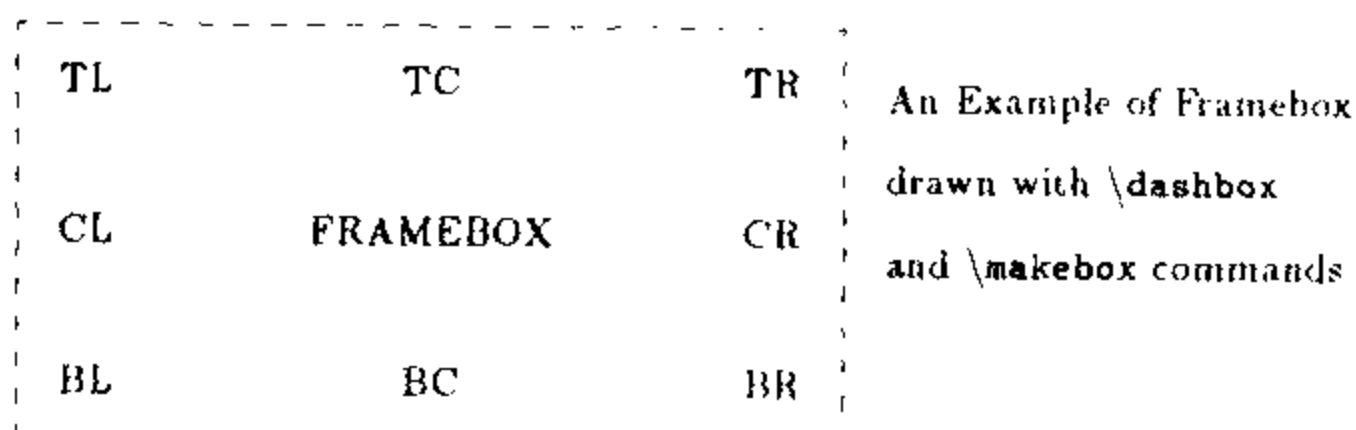


图 8.2 虚线矩形作图实例

读者不妨比较一下用命令“ $\backslash$ makebox”作为注解来“粘贴”文字和直接用“ $\backslash$ put”作图命令来将文字“画”在插图上的区别。

#### 四、实心矩形

实心矩形是指涂黑的矩形图形，一般用作描画较粗的尺子状线条、线框或各种尺寸的实心矩形。实心矩形的作图命令格式为：

```
\put(x,y){\rule{矩形宽度}{矩形高度}}
```

其中的“ $\backslash$ rule”注解命令的控制功能与LR排版状态下的功能完全相同(参阅本书第3.7.3小节)。

注意，矩形的宽度和高度值必须要带单位。为了与其它图形单位一致，用户也可采用作图基本单位“ $\backslash$ unitlength”。例如：

```

\unitlength=1mm
\begin{picture}(74.67,47.33)
\put(41.67,28.00){\rule{33.00\unitlength}{20.00\unitlength}}
\end{picture}

```

可画出一个长×宽为33mm×20mm的实心矩形。

### 8.3.3 画直线

在图形状态下, 画直线采用带参数控制命令“\line”来作为注解:

`\put(x,y){\line(Δx,Δy){Lx}}`

其中参数(Δx, Δy) 描述了需要绘制的直线斜率

$$k = \frac{\Delta y}{\Delta x}$$

$L_x$  表征了一般情况下( $\Delta x \neq 0$ ) 所画直线总长度在  $x$  方向的投影值的绝对值, 它间接地确定了直线的总长度。

说明:

① 系统读到画直线命令时, 会从参考点( $x, y$ ) 往点( $x + \Delta x, y + \Delta y$ ) 方向画直线。当  $L_x$  确定时, 如果  $\Delta x \neq 0$ , 则直线终点的坐标值为

$$(x_{\text{终点}}, y_{\text{终点}}) = \begin{cases} (x + L_x, y + L_x \frac{\Delta y}{\Delta x}) & \Delta x > 0 \\ (x - L_x, y - L_x \frac{\Delta y}{\Delta x}) & \Delta x < 0 \end{cases} \quad (8.1)$$

如果  $\Delta x = 0$ , 则终点的坐标值为( $x, y + L_x$ )。

② 由于L<sup>A</sup>T<sub>E</sub>X 只设置了一些确定的斜率供画直线时使用, 所以表征增量的参数(Δx, Δy) 只能取一些有限的数值。无论是Δx 或是Δy) 的取值范围都必须限制在整数区域[-6, 6] 之中(包括±6 在内); 两者对应的斜率值 $\frac{\Delta y}{\Delta x}$  必须是最简分数形式(例如(3, 6) 应改为(1, 2)); 此外, Δx 不能取“0”。注意下面例举的一些(Δx, Δy) 参数是不合法的:

(1.4, 3), (2, 4), (0, 3), (1, 8)

又例如当Δx = 0 时, Δy 只能取±1。

③ 虽然绘图环境picture 命令给出了作图盒子的大小, 但该命令所定义的作图区域对实际描绘各种图形时没有约束作用。换句话说, 你可能命将图形画出作图盒子的虚拟边界线。实际的排版效果是超出作图区域的部分可能会与正文重叠起来。

下列程序给出了较详细的一个图形排版状态下的画直线实例:

```
\begin{picture}(300,220)(0,0)
\put(100,0){\dashbox{5}(200,200){}}
\put(200,100){\line(1,0){100}}
\put(200,100){\line(5,1){100}}
\put(200,100){\line(4,3){100}}
\put(200,100){\line(1,1){100}}
\put(200,100){\line(3,5){60}}
\put(200,100){\line(2,5){40}}
\put(200,100){\line(1,5){20}}
\put(200,100){\line(0,1){100}}
\put(200,100){\line(-1,6){16.6}}
```

```

\put(200,100){\line(-1,2){50}}
\put(200,100){\line(-4,5){80}}
\put(200,100){\line(-5,4){100}}
\put(200,100){\line(-6,2){100}}
\put(200,100){\line(-4,1){100}}
\put(200,100){\line(-1,0){100}}
\put(200,100){\line(-3,-1){100}}
\put(200,100){\line(-5,-3){100}}
\put(200,100){\line(-6,-5){100}}
\put(200,100){\line(-3,-4){75}}
\put(200,100){\line(-2,-3){66.6}}
\put(200,100){\line(-1,-4){25}}
\put(200,100){\line(0,-1){100}}
\put(200,100){\line(1,-3){33.3}}
\put(200,100){\line(5,-6){53.3}}
\put(200,100){\line(3,-2){100}}
\put(200,100){\line(2,-1){100}}
\put(200,100){\line(6,-1){100}}
\end{picture}

```

输出结果如图8.3 所示。

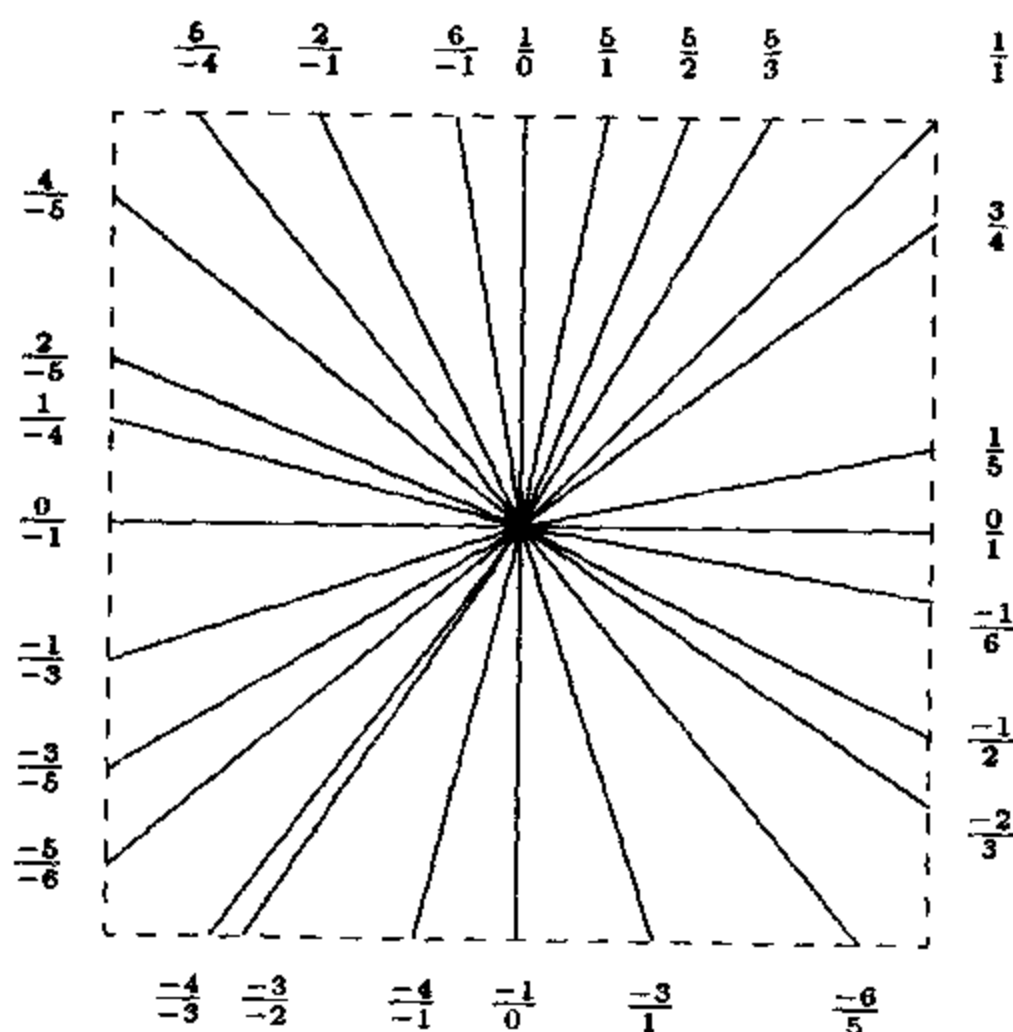


图 8.3 画直线示意图

图8.3 中在虚线矩形框周围标记各条直线的分数表示该直线的斜率，请读者对照排

版源程序, 分析一下每条直线线段从中央向四周射出的方向和线段在 $x$ 方向投影值的长度。

### 8.3.4 画箭头

画箭头的方法与画直线非常相似。在图形状态下, 画箭头用“矢量”(vector)控制命令作注解:

`\put(x,y){\vector{Δx,Δy}{Lr}}`

其中参数 $(\Delta x, \Delta y)$ 描述了需要绘制的箭头方向的斜率

$$k = \frac{\Delta y}{\Delta x}$$

$L_r$ 表征了通常情况下( $\Delta x \neq 0$ ), 所画箭头总长度在 $x$ 方向的投影值的绝对值, 它间接地确定了箭头的总长度。

说明:

① 系统读到画箭头命令时, 会从参考点 $(x, y)$ (箭头尾部) 往点 $(x + \Delta x, y + \Delta y)$ 方向画一条直线; 根据画箭头命令中指定的 $L_r$ 数值, 如果 $\Delta x \neq 0$ , 则箭头到顶端

$$(x_{\text{顶端}}, y_{\text{顶端}}) = \begin{cases} (x + L_r, y + L_r \frac{\Delta y}{\Delta x}) & \Delta x > 0 \\ (x - L_r, y - L_r \frac{\Delta y}{\Delta x}) & \Delta x < 0 \end{cases} \quad (8.2)$$

处停止, 并在该点绘上箭头记号。如果 $\Delta x = 0$ , 则箭头尖端点的坐标值为 $(x, y + L_r)$ 。

② 因为L<sup>A</sup>T<sub>E</sub>X只设置了一些确定的斜率供画箭头时使用, 所以表征增量的参数 $(\Delta x, \Delta y)$ 只能取一些有限的数值。与画直线命令不同的是 $\Delta x$ 和 $\Delta y$ 的取值范围限制在整数区域 $[-4, 4]$ 之间(包括 $\pm 4$ 在内); 两者对应的斜率值 $\frac{\Delta y}{\Delta x}$ 必须取最简分数式。

下面是一个在直角坐标系第一象限描绘矢量的源程序实例:

```
\begin{figure}[hbt]
\unitlength=0.7mm
\linethickness{0.4pt}
\begin{picture}(115.00,115.00)(-50,0)
\put(0.00,0.00){\vector(1,0){115.00}}
\put(0.00,0.00){\vector(0,1){115.00}}
\thicklines
\put(0.00,0.00){\vector(4,1){87.00}}
\put(0.00,0.00){\vector(3,1){86.00}}
\put(0.00,0.00){\vector(2,1){81.00}}
\put(0.00,0.00){\vector(3,2){76.33}}
\put(0.00,0.00){\vector(1,1){82.33}}
\put(0.00,0.00){\vector(4,3){72.67}}
\put(0.00,0.00){\vector(1,4){22.00}}
```

```

\put(0 00,0 00){\vector(1,3){28 33}}
\put(0 00,0 00){\vector(1,2){40 67}}
\put(0 00,0 00){\vector(2,3){49 67}}
\put(0 00,0 00){\vector(3,4){64 00}}
\put(106 33,-5 00){\makebox(0,0)[ct]{ $x$ }}
\put(-5 00,106 00){\makebox(0,0)[rc]{ $y$ }}
\put(-4 00,-4 00){\makebox(0,0)[rt]{ $o$ }}
\put(92 33,18 67){\makebox(0,0)[lt]{(4,1)}}
\put(91 67,28 67){\makebox(0,0)[lc]{(3,1)}}
\put(86 00,43 33){\makebox(0,0)[lc]{(2,1)}}
\put(80 00,52 33){\makebox(0,0)[lc]{(3,2)}}
\put(76 67,59 00){\makebox(0,0)[lb]{(4,3)}}
\put(66 67,66 00){\makebox(0,0)[lc]{(1,1)}}
\put(58 33,74 00){\makebox(0,0)[lc]{(3,4)}}
\put(52 33,79 00){\makebox(0,0)[lc]{(2,3)}}
\put(42 33,86 33){\makebox(0,0)[lb]{(1,2)}}
\put(30 00,90 33){\makebox(0,0)[lb]{(1,3)}}
\put(20 33,93 00){\makebox(0,0)[rb]{(1,4)}}
\put(0 00,0 00){\vector(1,0){91 00}}
\put(0 00,0 00){\vector(0,1){91 00}}
\put(90 00,4 67){\makebox(0,0)[rb]{(1,0)}}
\put(-4 00,86 33){\makebox(0,0)[rc]{(0,1)}}
\end{picture}
\end{figure}

```

输出结果如图8.4所示。图中显示的各矢量的参数是相应的方向数(即决定斜率的参数)。

### 8.3.5 堆积字符串

在许多插图中需要标记一系列有关图形各参数的说明文字或其它的字符串(类似单列矩阵或单列表格)。从单纯的排版角度来看,这可视作在由下向上堆积(叠加)字符串。使用 $\text{\LaTeX}$ 堆积字符串控制命令“ $\backslash\text{shortstack}$ ”作为注解,可以非常方便地编排成图形中成列状的文字。堆积字符串的命令格式为:

```
\put(x,y){\shortstack[对位参量]{字符串1\\字符串2\\...\\字符串n}}
```

说明:

① 此命令有点象无线表格环境(tabular)命令,它定义了一个独立的行文排版盒子,盒子的大小取决于盒子内的内容。盒子里边可编排一系列字符串,除了最后一个字符,各字符串间必须用控制字符“ $\backslash$ ”隔开。

② 参量 $(x,y)$ 是盒子的左下角坐标值,排版时,字符串 $n$ 处在盒子底部,而字符串1在顶部。

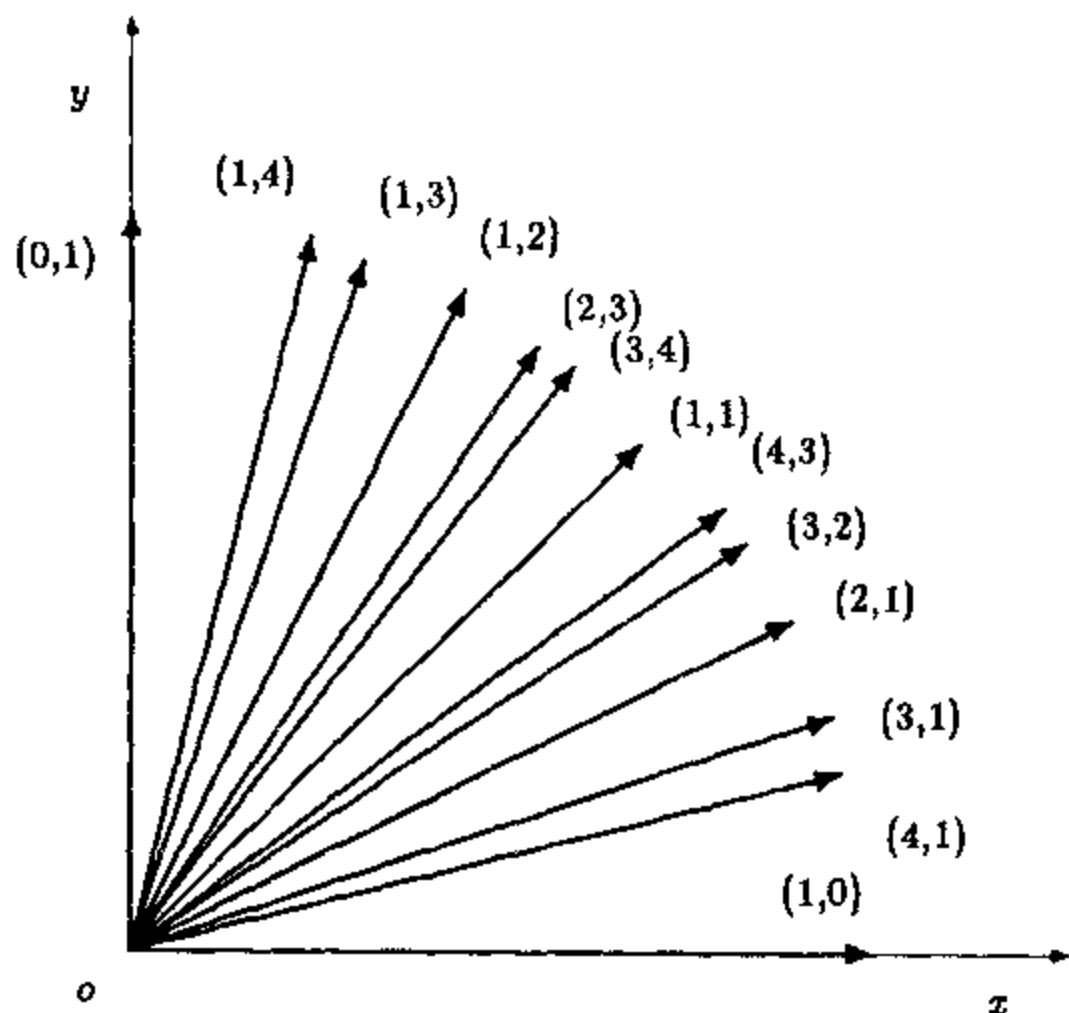


图 8.4 矢量(箭头)绘制示意图

③ 注解命令“`\shortatack`”可以在任意排版状态下使用。但是一般主要用在图形排版环境中。

下面是一段使用堆积字符串命令在一个图形中打印一系列文字的源程序实例:

```
\begin{picture}(200,106)(0,0)
\put(100,40){\shortatack[r]{
  $\stackrel{\normalsize\rightarrow}{\mbox{\bf A}}$ \, (2,1)$
  \\ $\stackrel{\normalsize\rightarrow}{\mbox{\bf B}}$ \, (2,3)$
  \\ $\stackrel{\normalsize\rightarrow}{\mbox{\bf A}}$+
  \stackrel{\normalsize\rightarrow}{\mbox{\bf B}} \, (4,4)$ }}
.....
\end{picture}
```

用户可以自己输入这段程序, 看看输出的结果如何。

### 8.3.6 画圆和圆弧

#### 一、空心圆

决定一个圆的几何条件只有两个: 圆心的坐标 $(x, y)$  和圆的直径 $d$ 。在图形排版状态下, 画空心源的L<sup>A</sup>T<sub>E</sub>X 控制命令为:

```
\put(x,y){\circle{d}}
```

注意, 由于各种新版本 $\text{\LaTeX}$ 提供可作图的最大圆直径 $d$ 的数值受到一定限制, 因此, 读者在使用时应查找有关参考手册, 或者用程序调试搜寻直径 $d$ 数值的上限(2.09 版本 $\text{\LaTeX}$ 系统可打印出的空心圆最大直径为40pt(点), 约14 毫米)。如果你设置的直径 $d$ 的数值大于系统容许的最大值, 则系统将你需要画的圆用直径最大值打印出来。至于圆上弧线的粗细仍由命令“ $\text{\thicklines}$ ”或“ $\text{\thinlines}$ ”来控制。

## 二、实心圆

画实心圆的 $\text{\LaTeX}$ 控制命令与画空心圆的命令没有多大差别, 只是在注解命令 $\text{\circle}$ 与后面的直径参数之间插入了一个星号“ $*$ ”作为实心圆控制符号:

```
\put(x,y){\circle*{d}}
```

排版时, 系统会在圆心位置为 $(x, y)$ 的地方绘制一个圆内部完全涂黑的直径为 $d$ 的实心圆。注意查询每个系统可绘制的最大圆直径(2.09 版本 $\text{\LaTeX}$ 系统可打印出的实心圆最大直径为15pt, 约5.3 毫米)。下面一个程序给出了分别绘制空心圆和实心圆的排版实例:

```
\begin{picture}(300,50)(0,0)
\put(204,37.5){\circle*{5}}
\put(204,12.5){\circle*{5}}
\put(191.5,25){\circle*{5}}
\put(216.5,25){\circle*{5}}
\put(204,25){\circle{40}}
\put(204,25){\circle*{3}}
\put(204,25){\circle{40}}
\put(204,25){\vector(1,1){14.1}}
\put(220,41){\makebox(0,0)[bl]{\$R\$}}
\end{picture}
```

输出结果如图8.5 所示。其中大圆半径 $R$ 为20pt(约7 毫米), 四个小实心圆的直径为5pt(约1.8 毫米), 而中央最小的实心圆的直径为3pt(约1 毫米)。

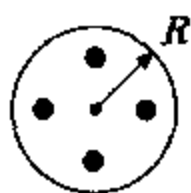


图 8.5 空心圆和实心圆绘制示意图

## 三、圆角矩形

有些椭圆状(oval)图形(如椭圆足球场等, 参见图8.6)可看成是将一个矩形的四个直角用合适半径的四分之一圆弧替换的结果, 称之为圆角矩形。 $\text{\LaTeX}$ 专门设计了一个圆

角矩形绘图命令:

```
\put(x,y){\oval(Lr,Lv)[绘制方式参量]}
```

其中参变量 $(x,y)$ 是圆角矩形对称中心的坐标值,而参数 $L_r$ 和 $L_v$ 分别是可容纳圆角矩形的长和宽。系统在读到此命令时,会自动为你寻找系统可能提供的合适的(与指定矩形的直线能光滑连接),且具有最大半径的四分之一圆弧来绘制图形。方括号内的绘制方式参量是用来对绘制全部图形还是局部图形作附加注解。绘制方式主要有两大类:第一类是选自参量“l”,“r”,“t”和“b”中的任意一个,所有输出的图形是整个圆角矩形的一半;第二类是选择前面四个参量中的某两个组成的另外四种绘制方式:“tl”,“tr”,“bl”和“br”,其特点是所绘制出的图形是整个圆角矩形的四分之一。具体的各个参量控制的输出方式可通过下面的程序实例

```
\begin{picture}(340,150)(0,0)
\put(200,75){\oval(65,25)}
\put(100,75){\oval(65,25)[l]}
\put(300,75){\oval(65,25)[r]}
\put(200,125){\oval(65,25)[t]}
\put(200,25){\oval(65,25)[b]}
\put(100,125){\oval(65,25)[tl]}
\put(300,125){\oval(65,25)[tr]}
\put(100,25){\oval(65,25)[bl]}
\put(300,25){\oval(65,25)[br]}
\end{picture}
```

以及用该源程序输出的实际拷贝(见图8.6)来理解。

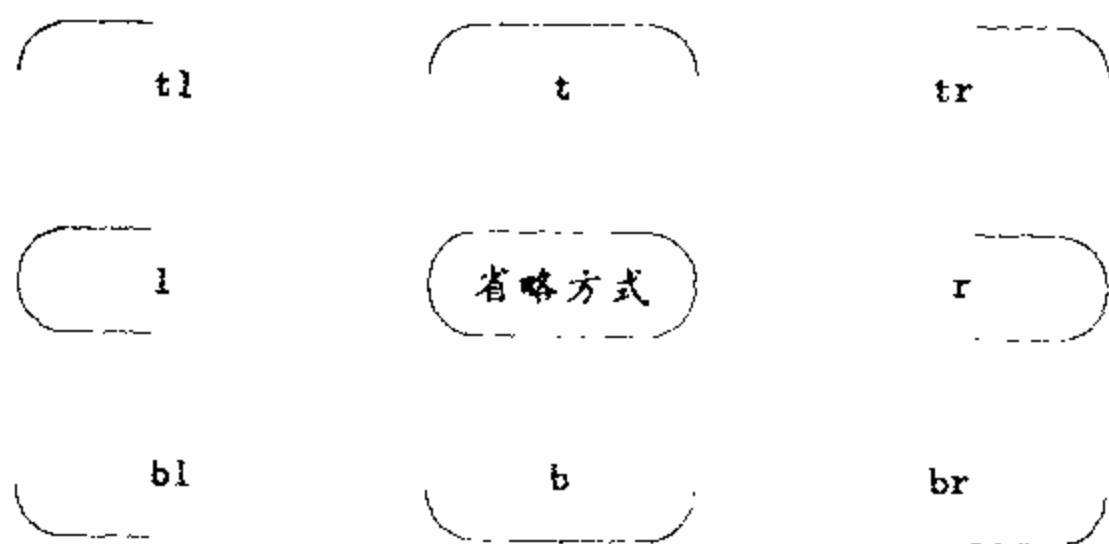


图 8.6 圆角矩形绘图方式示意图

如果用户需要将圆弧与确定的长直线或箭头光滑连接,则可将上述命令与特定的绘制直线或箭头的命令配合使用。例如程序:

```
\begin{picture}(340,50)(0,0)
```



```

\put(170,45){\line(0,-1){25}}
\put(180,20){\oval(20,20)[b1]}
\put(180,10){\vector(1,0){60}}
\end{picture}

```

可输出如图8.7所示的结果。

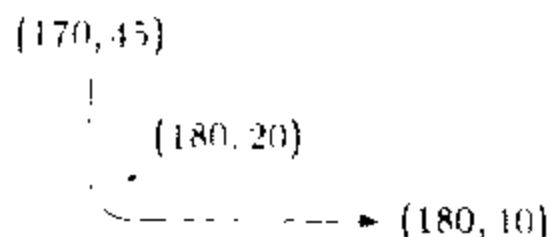


图 8.7 箭头或直线与圆弧光滑连接示意图

## 8.4 无线矩形图形的存取

由于图形排版状态下使用绘图命令作图所耗费的时间比较厉害, 所以采用在内存里存贮已经绘制好的图形, 并能让系统快速的调用就显得尤为重要。无线矩形图形的存取方法与在行文状态下存取无线框LR 排版盒子的控制令方法(参见第3.8 节)相比较, 两者的思路基本相同, 有些排版控制命令甚至完全一样。

### 8.4.1 图形存贮库的命名

要在系统内存里存贮一个在图形状态下绘制好的无线矩形图形(含有文字信息), 同样先要给存贮库命名, 以便以后调用时有一个图形专用名。可使用的给图形存贮库命名的控制命令形式与上一小节中介绍的给无线框LR 盒子存贮库命名的控制令完全相同:

```
\newsavebox{图形库名}
```

其中图形库名亦必须是由反斜杠控制符“\”与合法的字符串构成。

### 8.4.2 无线矩形图形的存贮

需要在已经命名好的存贮库里存贮一个含有文字信息的无线矩形图形, 可以使用命令:

```
\savebox{图形库名}(矩形宽度, 矩形高度)[对位参量]{注解文字}
```

这个控制命令中的选择参量与在图形排版状态下的无边界线框矩形图排版注解命令“\makebox”中的功能完全一样(见第8.3.2 小节)。执行排版时不会在命令所有的版面上打印出任何图形拷贝, 它会将绘制的图形存贮到已经命名的存贮库里。以后调用时, 系统不再进行具体计算作图, 而是直接将图形拷贝往插图上“粘贴”。

图形存贮命令也有一个无选择参量的简写式, 格式是:

```
\sbox{图形库名}{注解文字}
```

### 8.4.3 存贮图形的调用

对于源程序中处在图形排版状态下的任意坐标位置, 如果需要调用某个存贮库里的图形信息, 则可通过使用命令

```
\put(x,y){\usebox{图形库名}}
```

来直接获取存贮的无线矩形图形, 或者使用加线框的命令形式

```
\put{\frame{\usebox{图形库名}}}
```

获得实线矩形图。下面是一个存取图形的排版实例:

```
\newsavebox{\boxa}
\newsavebox{\boxc}
\begin{picture}(330,55)(0,0)
\sbox{\boxa}{\large \sf SBOX}
\savebox{\boxc}{50,25}[tr]{\sf SAVEBOX}
\put(104,38){\usebox{\boxa}}
\put(104,0){\usebox{\boxc}}
\put(104,0){\dashbox(50,25){}}
\put(179,38){\frame{\usebox{\boxa}}}
\put(179,0){\frame{\usebox{\boxc}}}
\put(254,38){\usebox{\boxa}}
\put(254,0){\usebox{\boxc}}
\sbox{\boxa}{}
\sbox{\boxc}{}
\end{picture}
```

输出结果如图8.8 所示:

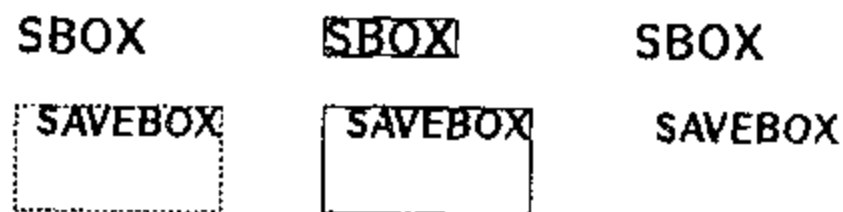


图 8.8 无线矩形图形的存取示意图

## 8.5 规则平移场合下图形的复制

在有些文稿中出现的一些规则图形里, 需要按简单数学规律线性地平移复制  $n$  个

相同的物体。如果只是按第8.3节介绍的基本的图形绘制命令来编写源程序, 必然会使小样文件变得比较冗长。

例如, 在图形状态下, 假定 $(x, y)$ 为版面上任意一点坐标值。现要求从该点出发, 往斜率为 $k = \frac{\Delta y}{\Delta x}$ 的直线方向上, 在坐标值依次为

$$(x, y), (x + \Delta x, y + \Delta y), (x + 2\Delta x, y + 2\Delta y), \dots, (x + (n-1)\Delta x, y + (n-1)\Delta y)$$

的位置上绘制 $n$ 个相同的图形。如果 $n$ 不太大, 也许你还能勉强输入 $n$ 条绘图控制命令

```
\put(x,y){图形注解}
\put(x+\Delta x,y+\Delta y){图形注解}
\put(x+2\Delta x,y+2\Delta y){图形注解}
.....
\put(x+(n-1)\Delta x,y+(n-1)\Delta y){图形注解}
```

但是假如 $n$ 数值很大, 程序就显得太长了。LaTeX 充分利用计算机可作循环工作的特点, 对上述情形设计了一个适用于规则平移场合下的多次图形复制命令:

```
\multiput(x,y)(\Delta x,\Delta y){n}{图形注解}
```

这样, 一条简单的语句就能发挥出与上面 $n$ 条绘图控制命令相同的作用。例如程序:

```
\begin{picture}(340,65)(0,0)
\multiput(104,50)(20,-5){11}{\vector(1,1){15}}
\end{picture}
```

可输出如图8.9所示在一个方向上绘制11个箭头的结果。

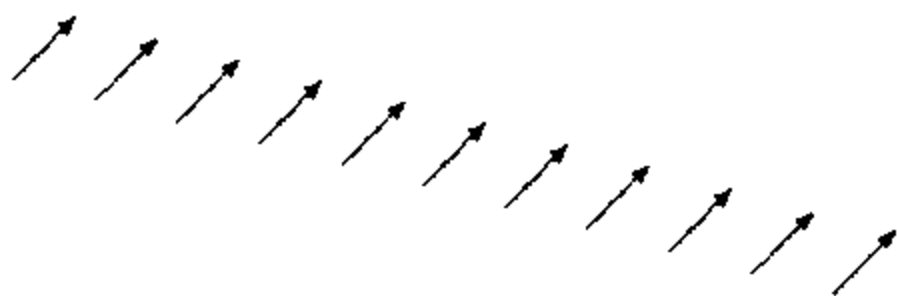


图 8.9 箭头线性平移复制示意图

如果在图形注解命令中再次嵌套图形环境命令, 并仍然调用“\multiput”, 则还可轻而易举地获得在两维情形下的复制图形。例如程序:

```
\begin{picture}(320,75)(0,0)
\multiput(104,50)(10,0){21}
{ \begin{picture}(0,0)(0,0)
\multiput(0,0)(0,-10){5}{\circle*{2}}
\end{picture} }
\end{picture}
```

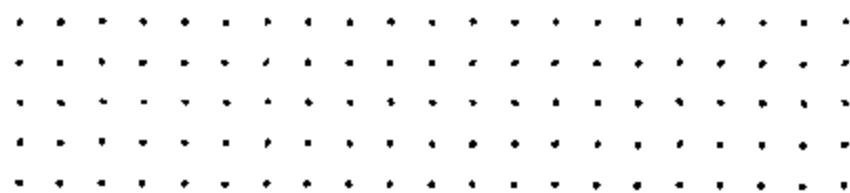


图 8.10 实心圆两维复制示意图

可以得到如图8.10所示的两维实心圆阵列。

注意，由于循环语句会占用一定内存，所以在两重循环中，应尽量避免占用太多的内存空间而导致系统运行存储空间危机(例如计算机有时会出现拒绝执行控制命令而自动退出系统的异常现象，甚至会出现突然“死机”<sup>1</sup>状况)。

---

<sup>1</sup>“死机”是指非正常的紧急停机，此时，计算机无法再作任何工作。遇到这种情况，你只能通过热启动或冷启动来重新恢复MS-DOS操作系统。

## 9

# 交叉引用

本章介绍如何在文件的章节、方程、图表、文献和索引等交叉引用过程中实现智能化。

在一本普通教科书或者学术专著中总会牵涉到交叉引用问题。所谓交叉引用,是指在正文中的某段文字中需要引用在前面已经叙述过的(也可能是在后面将会附录的)某些内容。这些内容,可以是泛指某一章某一节,也可以是某个图或表,或者是某个方程、某个参考文献,等等。它们的特点是均可以按各自的内容范畴归类编号。例如对某篇文章中的第三个图和第四个表格,在引用时可以分别称之为图3和表4;同样的对假定在前面正文中出现过的第五个数学方程式和文章后面附录中的第六篇参考文献,在再次引用时可分别简称之为方程(5)和文献[6]。

现在假定用户要编排一篇频繁地出现交叉引用的学术论文。如果文稿已经定型,不再改变,那么无论内容怎么复杂,实现电脑排版并不是一件十分困难的事,需要的只是耐心和仔细而已。但是问题是所谓“定型”和“不变”,只要是还没有最后付印,对一个治学非常严谨的(特别是对自己的作品极其挑剔的)作者来说,通常是不可能的。当一篇文章的雏形问世到出版商最后大规模印刷,往往会经历许多次修改。这种修改,有时只是简单的文字修饰,这对电脑排版来说,只是区区小事;但是有时是大刀阔斧的版面修改,例如内容的增、删,文章前后布局的调整等,这对排版程序员来说就非常头痛了。

例如,应审稿者的要求,某个作者在他的文章中必须增加一篇参考文献和删去一个数学方程式。对程序员来说,这意味着在附录中的从增添的参考文献位置起后面出现的文献编号均要增加数值“1”,而在文章中所有引用的相应文献的编号亦需全部作更改,否则就会出现“张冠李戴”的笑话;同样的在被删除的数学方程式以后出现的所有方程编号和在文章中引用的相应的方程代号都要依次减去数字“1”,否则的话会造成严重错误。

两个小小的改动,就会令程序员头昏眼花,而且不一定就不犯错,因为如果一本书中引有几十篇参考文献和上百个数学方程式,那么在文章排版的源程序中寻遍所引文献或方程编号就不那么容易了。现在问题是,如果文章更改的地方很多,那又会怎

样呢,想必没有人愿意跟在挑剔的作者后面老是做一些异常乏味的编号数字的更改工作吧!

将排版系统智能化,使打字员从呆板的数字更改工作中解放出来,这是所有排版系统设计者的一个重要奋斗目标。迄今为止,在诞生的国内外诸多排版系统中,实现交叉引用智能化最强的,当首推 $\text{\LaTeX}$ 了。 $\text{\LaTeX}$ 排版系统不仅会自动为用户所排文件中的章节、方程、图表和文献索引等编号,而且在用户为每一个项目(将来会引用的章节、方程、图表和文献索引等)设定关键词后,可以准确无误的实现交叉引用。无论作者作多大规模的修改,或删除,或增添,所有的交叉引用过程中的繁冗数字更改会在系统执行排版的瞬息过程中自动实现,而且绝对不会出错。

## 9.1 章节图表目录

### 9.1.1 章节目录的生成

全部文件中的完整的章节标题在目录中的引用非常方便。如同在第五章第五节列目录命令中介绍的,只要用户是按照在本书第五章中叙述的几个标准的版式命令去编辑源文件的话,那么系统在对文件执行排版时就会自动记录下所有关于各章节的目录的完整信息(包括对应的页码和标题)。当用户需要目录时,只要在适当位置输入命令:

```
\tableofcontents
```

$\text{\LaTeX}$ 在每次读到此命令时,就会为用户做如下两件事:

① 读入扩展名为[.TOC]的老版本文件,根据输入的有关章节内容和格式的信息编排目录,并自动加上黑体字英文目录标题“Contents”。注意在第一次运行的时候,若无老版本的TOC文件,则产生一个仅有标题为“Contents”的空白版面。

② 命令 $\text{\LaTeX}$ 在计算机硬盘里建立产生一个新版本的TOC文件,其引用名与 $\text{\LaTeX}$ 小样TEX文件相同。在 $\text{\LaTeX}$ 对小样文件的正文部分执行排版时,会依次记录下有关新版本目录中包含的各章节内容的有关信息。

### 9.1.2 图表目录的生成

如果你想在章节目录后面列出有关文件中图表的信息,那么可在列目录命令后面再分别输入“列表目录命令”

```
\listoftables
```

和“列图目录命令”

```
\listoffigures
```

这两个命令的功能与与列目录命令相似,只是产生的新版本的目录信息文件的扩展名不同。对列表目录命令将产生扩展名为[.LOT]的文件,而对列图目录命令则将产生扩展名为[.LOF]的文件。

如果用户不喜欢目录文件为你提供的目录格式，那么在确定最后版本的文件输出时可以对有关的目录信息文件进行适当修改。不过用户得十分小心。有关这方面的知识可查阅L. Lamport 书《 $\text{\LaTeX}$ : A Document Preparation System》

## 9.2 章节图表的交叉引用

### 9.2.1 引用标识符

要运用自如、且准确无误地实现文件中的章节、方程、图表和文献索引等的交叉引用，关键的是要在所要引用的部位设置供以后调用的标识符(key)——或称之为关键词。然后由 $\text{\LaTeX}$ 为你将这些标识符根据所处的环境或位置翻译成相应的章节编号，或者图表编号等。在以后调用时，只要提到所需引用部分的标识符，就能非常方便地实现交叉引用。

引用标识符的设置方法非常简单，只要你在所要引用的任何部位插入引用标识符命令：

`\label{关键词}`

引用标识符命令可以是设在某一章某一节内，也可以是设在某个作图或制表环境内，或者插在某个编号数学方程式后面。命令中标识用的关键词可以是任意的大小写的西文字符串，也可以是数字或标点符号。注意西文字符中的大小写可以标识不同的对象。在 $\text{\LaTeX}$ 读到此命令时，就会在AUX信息文件中记录下关键词所对应的标号。

在需要引用的时候，用户只要在确定的位置插入交叉引用命令：

`\ref{关键词}`

就可以实现交叉引用。在 $\text{\LaTeX}$ 读到交叉引用命令时，就会在AUX信息文件中搜寻该关键词所对应的标号。提请读者注意的是：与列目录命令相同，在你第一次让 $\text{\LaTeX}$ 为你执行排版时，由于系统中还没有在硬盘中建立AUX文件，你所引用的标识符也就没有任何参考对象。此时，系统会向你报警：“你所引用的标识用关键词未被定义过”。如果你重新再运行一遍 $\text{\LaTeX}$ ，问题即可解决。

可以实现交叉引用的排版环境主要有：

- ① 章节内容中的任何位置，例如在命令“`\chapter`”、“`\section`”和“`\subsection`”后面；
- ② 标号数学方程式中的任何部位，例如在单行标号方程式环境命令(`equation`)或多行标号方程组环境命令(`eqnarray`)中；
- ③ 任何图(`figure`)表(`table`)环境命令之中；
- ④ 在任何标号列条目环境(`enumerate`)之中。

注意：脚注版面与其它版面不同，脚注的正文内容只能在所处页面内用脚注命令来引用。用户不能采用标识符方法来进行交叉引用。

### 9.2.2 章节的交叉引用

假如某一章节的内容在其它章节中被引用时是泛指整章或整节, 那么为了查寻方便, 引用标识符通常就直接设置在该章或该节的标题命令下面。例如在一篇书籍类(book)文件格式的小样中, 有如下一段排版程序:

```
... From Section \ref{sec-usetex} you will get to know
how use \LaTeX.
.....
\section{Running \LaTeX}
\label{sec-usetex}
Let's say you have the following input file, named
{\tt mydoc.tex}:
.....
```

那么在排版输出时可以得到:

From Section 2.3 you will get to know how use  $\text{\LaTeX}$ .

:

#### 2.3 Running $\text{\LaTeX}$

Let's say you have the following input file, named mydoc.tex:

.....

如果在第2.3节前面增添了一节, 那么系统会自动令在标题前的标号以及所引用的章节编号增添数字“1”。有了标识符命令中的关键词“sec-usetex”, 你再也不用去理会如何更改标号的麻烦事了!

### 9.2.3 方程的交叉引用

数学方程式的引用标识符可以设置在该数学方程式的排版环境命令之中, 一般就安排在数学环境结束命令之前(例如在命令“\end{equation}”前面), 即整个方程式内容的最后面。例如:

```
Newton's second law used for a system of
particles gives the formula
\begin{equation}
\mbox{\bf F}_{\mbox{ext}}=\frac{\mbox{d}\mbox{\bf P}}{\rm d}t
\label{eq:Newton}
\end{equation}
.....
Suppose that the sum of the external forces acting
on a system is zero. Then, from Eq. (\ref{eq:Newton}),
we obtain the {\em Principle of Conservation of
```



```

Linear Momentum}.
\begin{equation}
\mbox{\bf P} = \mbox{a constant vector}
\end{equation}
.....

```

输出结果:

Newton's second law used for a system of particles gives the formula

$$\mathbf{F}_{\text{ext}} = \frac{d\mathbf{P}}{dt} \quad (9.1)$$

.....

Suppose that the sum of the external forces acting on a system is zero. Then, from Eq.(9.1), we obtain the *Principle of Conservation of Linear Momentum*.

$$\mathbf{P} = \text{a constant vector} \quad (9.2)$$

.....

#### 9.2.4 图表的交叉引用

如果作图或制表环境中的标题注文(caption)只有一个,那么图表的交叉引用标识符既可以设置在标题注文参量之中,也可以设置在图形或制表环境命令中的任意位置。但是假如某个环境中的标题注文(caption)有若干个,那么标识符必须设置在所引用的表图注文命令后面,或者插入表图标题命令的注文参量中间。

#### 9.2.5 页面的交叉引用

页面的交叉引用是指特定的某一页内容。假定用户在文稿的某个将要被引用的特定位置标记了引用标识符,按通常约定,在使用命令“\ref”时,只会得到所在章节的序号信息。如果你对章节序号信息不感兴趣,而只注重引用内容所处的页码(便于读者迅速查阅),那么可使用页面引用命令:

```
\pageref{关键词}
```

例如程序:

```

See page~\pageref{'energy'} for more details.
.....
$E$ is called the total mechanical
energy \label{'energy'}

```

会在某一版面上输出:

See page 139 for more details.

⋮

而在第139页上, 你会发现:

...  $E$  is called the total mechanical energy.....

由于引用的关键是编号数字, 用户在具体使用时可在源程序中添加“page”, “equation”和“section”等。为了防止这些单词与后面引用的编号断开(如遇到分行和换页), 通常使用控制符“~”将前后连结起来。

## 9.3 参考文献

科技学术论文中的参考文献(bibliography)是指那些由作者在文章中指明作品中的论点或数据出处的索引资料, 或者是向读者推荐的可供参阅的文献篇目。按国际惯例, 参考文献通常附于文末。为了防止与方程式中的注释号(1), (2), (3), ...相混淆, 作为参考文献序码的阿拉伯数字一般置于方括号内。对于排版系统而言, 参考文献就是交叉引用的一个重要特例。

### 9.3.1 参考文献的编排

L<sup>A</sup>T<sub>E</sub>X系统中参考文献的编排是设置在专门的排版环境中进行的, 它的标准排版环境的命令格式是:

```
\begin{thebibliography}{最长的序码字符串}
\bibitem[序码1]{关键词1} <文献1正文>
\bibitem[序码2]{关键词2} <文献2正文>
.....
\bibitem[序码n]{关键词n} <文献n正文>
\end{thebibliography}
```

说明:

① 在每条参考文献的正文最前面的方括号中的序码*i*是指用户在自己选取的计数方式中的序码字符串。你可以选用字母a, b, c, ..., 来作为文献的序号, 也可以使用罗马数字, 还可以用其它任意形式的字符串。这部分内容可省略。省略时系统将用阿拉伯数字为文献编号。

② 参数最长的序码字符串 是要求用户指明的在文献编排时所采用的序码形式中可能出现的最长的字符串。它的功能是控制在编排每条参考文献的正文时所留给序码排版的空距大小。如果用户未采用自己的序号编码形式, 参数值可选取可能出现最大的阿拉伯数字。例如, 对于文末附有几十条参考文献的情形, 参数最长的序码字符串 可设置为最大的两位数“99”。

③ 标识每条文献的关键词*i*的功能与前面提到的引用标识符命令中设置的关键词功能相同。这里只是省去了控制令“\label”而已。

④ 排版时,对文件格式参量选用了文章类(article)时,系统在排版后会在参考文献的最前面加上黑体字标题“References”,面对报告类或书籍类文件则会加上标题“Bibliography”。

### 9.3.2 参考文献的引用

参考文献的引用是通过命令:

`\cite{关键词i, 关键词j, 关键词k,...}`

来实现的。在文稿中出现引用文献时,系统对每次引用的数目多少不予限制,用户只要将相应的若干个标识参考文献的关键词填进命令“\cite”后面的花括号内就可以了。注意:由于空格亦可当作标识字符,所以在用逗号“,”隔开的参量关键词之间不要留空格。以下是一个在文章类文件格式中的参考文献的排版实例:

```
... See Refs.~\cite{Feng,Liu} for the hairy details.
.....
\begin{thebibliography}{99}
.....
\bibitem{Feng} W. FENG, {\em et al.},
{\em Phys. Rev. B} {\bf 43} (1991) 6893.
.....
\bibitem{Liu} N. Liu, {\em et al.},
{\em J. Phys.: Condensed Matter} {\bf 4} (1992) 9823.
.....
```

输出时,你可以发现如下结果:

```
.... See Refs. [58,98] for the hairy details.
:
(在文章末尾处)
References
:
[58] W. FENG, et al., Phys. Rev. B 43 (1991) 6893.
:
[98] N. Liu, et al., J. Phys.: Condensed Matter 4 (1992) 9823.
:
```

如果你在正文中引用文献时,需要在填写序码的方括号内再添加一些注文,那么还可以在引用文献的控制命令中插入一个注文参量,

`\cite[文献注文]{关键词1, 关键词2, 关键词3, ...}`

例如小样

```
See Ref.~\cite{Feng} or Ref.~\cite[pages 9823--9830]{Liu} for
information on the editing of \LaTeX{} files ...
```

输出时,  $\text{\LaTeX}$  会为你将文献注文参量“pages 9823 9830”用逗号“,”隔开后添加在文献序码后面:

```
See Ref. [58] or Ref. [98,pages 9823 9830] for information on the editing of \LaTeX
files ...
```

如果你不喜欢上面用阿拉伯数字计数的序码形式, 那么可以将源程序改为:

```
... See Refs.~\cite{Feng,Liu} for the hairy details.
.....
\begin{thebibliography}{zz}
.....
\bibitem[ab]{Feng} W. FENG, {\em et al.},
{\em Phys. Rev. B} {\bf 43} (1991) 6893.
.....
\bibitem[ba]{Liu} N. Liu, {\em et al.},
{\em J. Phys.: Condensed Matter} {\bf 4} (1992) 9823.
.....
```

输出时, 你可以得到用标签“ab”和“ba”等来作为序码的结果:

```
.... See Refs. [ab,ba] for the hairy details.
```

```
:
```

(在文章末尾处)

## References

```
:
```

```
[ab] W. FENG, et al., Phys. Rev. B 43 (1991) 6893.
```

```
:
```

```
[ba] N. Liu, et al., J. Phys.: Condensed Matter 4 (1992) 9823.
```

```
:
```

## 9.4 编制索引

无论是对作者、编辑或是排字员, 编制一个索引并不是一件容易的事情。使用 $\text{\LaTeX}$ 排版系统, 就能使你想要在一篇较长文章或很厚的书籍后面编制一个详细的索引成为可能。 $\text{\LaTeX}$ 编制索引是分两步进行的。首先是在文件正文中所有需要列入索引的项目

所处的位置“贴上标签”,让系统在排版过程中生成一个索引信息文件。然后根据索引信息编制一个索引。

### 9.4.1 生成索引信息文件

所谓索引信息,是指文件正文中需要罗列的所有条目及其在排版以后每个条目所对应的版面页码。生成索引信息文件必须要在文稿排版环境之前(在文件格式(document-style)命令和文稿(document)排版环境起始命令之间)加进一条产生索引文件的控制命令:

```
\makeindex
```

LaTeX 在读到此命令时立即会在硬盘中建立起一个扩展名为[.IDX]文件(引用名与TEX文件(小样排版文件)相同的标准ASCII文本文件)。

对于文稿正文内需要列入索引中的每个项目,用户可以通过在紧靠该项目后面插入一个带单参量的索引标签命令来实现在IDX文件中的逐项信息记录。索引标签命令的具体形式为:

```
\index{条目标题}
```

例如源程序

```
\documentstyle{book}
.....
\makeindex
.....
\begin{document}
.....
The motion of a given particle\index{particle} is
determined by the nature\index{nature} and the
arrangement of the other bodies that forms its
environment\index{environment}.
.....
```

在系统执行排版后会在IDX文件中作如下记录(假定上面一段文稿在输出时恰被排在第42页和43页之间):

```
:
\indexentry{particle}{42}
\indexentry{nature}{42}
\indexentry{environment}{43}
:
```

注意不要在命令`\index`和所要注解的条目之间留任何空格。因为万一跨页现象出现的话,你在IDX文件中获得的页码也是错的(数字会增加1)。当你在源文件中“粘”上了所有“标签”之后,运行`LATEX`即可获得完整的IDX文件。

注意:

- ① 如果没有在源文件之前设置命令`\makeindex`,则所有“标签”无任何作用。
- ② 命令`\index`中的条目标题参量可以是任意字符串包括几个特殊的控制符,如“`\`”和“`$`”,不过应恰当地设置好分组控制符“`{}`”。
- ③ 不要将命令`\index`作为参量嵌套在其它命令中。

假如你已经产生了IDX文件,那么下面的工作是如何由IDX文件中的有关信息来编制完整的文件索引。

### 9.4.2 编制文件索引

文件索引的编制必须是在特定的索引(`theindex`)排版环境中进行,每个条目的排版是通过列条目命令(`\item`, `\subitem`, `\subsubitem`,...)来设置的。至于条目之间的纵向空距则要靠专用的索引纵向空距命令“`\indexspace`”来调整。在系统对索引部分的小样执行排版时,会根据国际印刷业惯例按双栏版式排版(与正文部分设置的单双栏的格式无关),并且会在索引正文的最前面自动为你印出黑体字标题“**Index**”。例如小样文件

```
\begin{theindex}
.....
\item particle dynamics 1, 15
  \subitem environment 43, 159
  \subitem mass 10, 55
    \subsubitem center mass 60
  \subitem nature 42, 78
  \subitem particle 42, 55, 65
\indexspace
\item work and energy 103, 159
.....
\end{theindex}
```

输出结果为

```

:
Index
:
particle dynamics 1, 15
  environment 43, 159
  mass 10, 55
```

center mass 60  
nature 42, 78  
particle 42, 55, 65  
  
work and energy 103, 159  
:

## 9.5 编制词汇表

编制一个词汇表(glossary)方法与编制索引的方法完全类似。只不过将对应的命令改换名字而已(使用本节命令之前应先阅读前面一节内容):

① 词汇标签命令 取代带单参量索引标签命令“\index”的是命令“\glossary”, 后面携带的参量的选用规则与索引标签命令完全一样。

② 词汇信息文件生成命令 命令“\makeglossary”可为你生成一个扩展名为[.GLO]的词汇信息文件。在文件中的每个记录均由“\glossaryentry”来引出。其参数意义与索引信息文件中的“\indexentry”没有区别。





## 10

# L<sup>A</sup>T<sub>E</sub>X 程序设计高级技巧

本章介绍L<sup>A</sup>T<sub>E</sub>X排版系统中一些比较实用的程序设计高级技巧,例如自行定义一个L<sup>A</sup>T<sub>E</sub>X控制命令,在执行排版过程中实现人机对话,在一个L<sup>A</sup>T<sub>E</sub>X源文件中调用另外一个源文件,改变系统的计数(页码、章节编号等)方式,自行设计列表环境,调用排版系统配备的其它新字库,帮助系统进行断字、断行和断页,等等。

### 10.1 自行定义L<sup>A</sup>T<sub>E</sub>X控制命令

在排版过程中,往往会遇到一小块特殊版面的复杂内容(例如分式、矩阵方程根式等)被重复编排了多次。如果将此种类型的版面通过一个排版子过程控制命令事先定义好,而以后需要时只要在源程序中直接调用该排版命令,这对用户来说,可以大大节省源程序的键盘输入时间。此外,有些反复使用的较长的排版控制命令,由于命令注解词较长(例如脚注字号命令“\footnotesize”)输入多次必使人感觉厌烦。若能自行简化(例如将注解词缩短),则会减轻一些用户在输入时的疲劳感觉。考虑到以上因素,L<sup>A</sup>T<sub>E</sub>X语言中提供了允许用户自行定义排版控制命令的有效编程语句。

#### 10.1.1 定义一个排版子过程控制命令

自行定义一个L<sup>A</sup>T<sub>E</sub>X排版子过程控制命令(不带参量)的标准格式为:

```
\newcommand{自定义控制命令}{排版子过程程序}
```

其中参量自定义控制命令必须是以反斜杠转码符“\”打头的英文字符串,而排版子过程程序则是标准的L<sup>A</sup>T<sub>E</sub>X源程序,可以是西文单词、短文、控制命令或数学公式等,也可以是已经定义好的控制命令。例如在下面程序中定义了若干个排版子过程的控制命令:

```
\newcommand{\B}{\mbox{\bf B}}
```

```

\newcommand{\dr}{\mbox{d\bf r}}
\newcommand{\integralfunc}{\mbox{\B$\cdot$\dr}}
\newcommand{\lineintegral}{\mbox{$\oint \B \cdot \dr$}}
\newcommand{\proj}{\mbox{projection}}
.....
We wish to evaluate \lineintegral\ for two closed
paths---a path that surrounds the wire and one does not.
We can view \integralfunc\ as the product of $\B$ and
the \proj\ of \dr\ onto \B. Because the field \B\
around straight wire has only an azimuthal component,
the \proj\ of \dr\ onto \B\ is $r\,\mbox{\rm d}\phi$ for either
path. Therefore, we have
\begin{equation}
\oint \integralfunc = \oint B(r)r\,\mathrm{d}\phi
\end{equation}

```

输出结果为:

```

.....
We wish to evaluate  $\oint \mathbf{B} \cdot d\mathbf{r}$  for two closed paths—a path that surrounds the wire
and one does not. We can view  $\mathbf{B} \cdot d\mathbf{r}$  as the product of  $B$  and the projection of  $d\mathbf{r}$ 
onto  $\mathbf{B}$ . Because the field  $\mathbf{B}$  around straight wire has only an azimuthal component,
the projection of  $d\mathbf{r}$  onto  $\mathbf{B}$  is  $r d\phi$  for either path. Therefore, we have

```

$$\oint \mathbf{B} \cdot d\mathbf{r} = \oint B(r)r d\phi \quad (10.1)$$

说明:

① 作为控制命令的字符串不能含有阿拉伯数字或其它非英文字母的字符, 例如以下定义的控制命令是不合法的:

```

\r12      \integtal2      \func1a      \#Number

```

此外, 自行定义的控制命令的名字也不能与已有的L<sup>A</sup>T<sub>E</sub>X 控制命令或已经定义了的控制命令同名, 否则在执行排版时系统会向用户报错。

② 自定义的控制命令可以放在源程序中的任意位置, 但必须是位于第一次调用该命令的程序语句之前。为了便于查阅, 建议将新控制命令的定义过程放在全局说明区域内。

③ 自定义的控制命令可以与其它L<sup>A</sup>T<sub>E</sub>X 命令或已经定义好的控制命令嵌套使用。但是在定义语句中不能嵌套另一个含有“\newcommand”的控制命令。

### 10.1.2 含变量控制命令的自定义方法

L<sup>A</sup>T<sub>E</sub>X 系统的许多排版控制命令是含参变量的。自行设计类似的带可调参量的控

制命令需要将排版子过程控制命令的定义方法作些扩充。若要定义一个带 $n$ 个参量的 $\text{\LaTeX}$ 排版子过程控制命令，可按如下格式来设置：

```
\newcommand{自定义控制命令}[n]{... #1 ... #2 ..... #n ... }
```

说明：

① 方括号内的参数“ $n$ ”是用来确定自定义控制命令所含参数的个数，“ $\#i$ ”表示嵌在排版子过程中的虚拟的第 $i$ 个参变量，而“...”则表示其它固定的辅助排版过程。

② 嵌在排版子过程中的 $n$ 个虚拟参变量的顺序可以打乱，并且各个虚拟参变量在自定义的排版子过程中可以重复使用。

③ 在调用自定义的含参变量控制命令时，用户可将实际输入的 $n$ 个参数用 $n$ 对花括号括起来后置于自定义控制命令的后面。注意，取代虚拟变量的实际参量可以是任意排版语句。

④ 为了减少引进的参变量中可能出现的排版控制命令对其它定义的排版子过程产生不必要的影响，建议将虚拟参变量符号“ $\#i$ ”两边增添一对分组花括号“ $\{\}$ ”。

例如程序

```
\newcommand{\wavefunc}[3]{\mbox{$\Psi(\{#1\},\{#2\};\{#3\}) =  
\exp [i\{#3\}\cdot\{#1\}-\omega\{#2\}]\$}}
```

定义了一个出现在量子力学教程中的波函数。这个函数在以下排版实例中被连续调用了两次。

```
\newcommand{\fx}[2]{\mbox{(\bf #1$_{#2}$)}}  
\newcommand{\rone}{\fx{r}{1}}  
\newcommand{\rtwo}{\fx{r}{2}}  
\newcommand{\pone}{\fx{p}{1}}  
\newcommand{\ptwo}{\fx{p}{2}}  
.....  
Suppose we have two {\em plane waves} for the two particles  
with momentum \fx{p}{i}, and at position \fx{r}{i} ($i=1, 2$)  
respectively:  
\begin{equation}  
\wavefunc{\rone}{t}{\pone}  
\end{equation}  
\begin{equation}  
\wavefunc{\rtwo}{t}{\ptwo}  
\end{equation}
```

在对此源文件执行排版后，你将获得：

.....  
Suppose we have two *plane waves* for the two particles with momentum  $p_i$ , and at

position  $\mathbf{r}_i$  ( $i = 1, 2$ ) respectively:

$$\Psi(\mathbf{r}_1, t; \mathbf{p}_1) = \exp[i(\mathbf{p}_1 \cdot \mathbf{r}_1 - \omega t)] \quad (10.2)$$

$$\Psi(\mathbf{r}_2, t; \mathbf{p}_2) = \exp[i(\mathbf{p}_2 \cdot \mathbf{r}_2 - \omega t)] \quad (10.3)$$

注意, 在源程序中另外又定义了一个带双参量的控制命令“\fx”, 并且被其它控制命令多次调用。

### 10.1.3 重新定义已经设置的控制命令

L<sup>A</sup>T<sub>E</sub>X 系统已经为用户定义了许多具有特殊排版功能的控制命令。如果需要, 用户还可以自行设计一些控制命令。但是自定义的控制命令绝对不能与系统保留的或已经自行定义的控制命令同名。如果出现同名, 则会产生排版错误。然而, 有时用户不得不将现有的 L<sup>A</sup>T<sub>E</sub>X 控制命令的排版功能作某些修改, 这就需要对原有的命令重新定义。L<sup>A</sup>T<sub>E</sub>X 系统中配置了对现存的控制命令的排版功能重新定义的编程语句: 对无参量的控制命令, 重新定义的语句格式为

`\renewcommand{已定义的控制命令}{排版功能注解程序}`

而对带参量的控制命令, 其格式为

`\renewcommand{已定义的控制命令}[n]{... #1 ... #2 ... #n ... }`

说明:

① 控制命令“\renewcommand”的使用方法与“\newcommand”完全类似, 只是定义的对象不同而已。“\renewcommand”是对已经存在的控制命令作排版功能的修改, 而

`\newcommand`

则是定义一个新的排版命令。

② 如果用户没有弄清楚自己究竟在干什么, 千万不要轻易修改 L<sup>A</sup>T<sub>E</sub>X 系统设置的标准命令。不然的话, 你也许不能获得意想之中的排版效果。

③ 重新定义的控制命令可以与其它 L<sup>A</sup>T<sub>E</sub>X 命令或已经定义好的控制命令嵌套使用。但是在控制命令重新定义语句中不能嵌套另一个含有“\newcommand”或“\renewcommand”的控制命令。

## 10.2 自行定义 L<sup>A</sup>T<sub>E</sub>X 排版环境

### 10.2.1 排版环境控制命令的化简

有些使用频率较高的排版环境控制命令, 由于环境注解词较长(如条目列表环境命令注解词“enumerate”), 重复输入亦会使人感到厌倦。倘若能将环境注解词简化, 则会大大减轻用户在输入程序时的疲劳感觉。因为排版环境控制命令是由起始和结束两个

命令构成的, 所以要想将 $\text{\LaTeX}$ 的排版环境控制命令化简的话, 必须配对进行。具体步骤如下:

```
\newcommand{自定义环境起始控制命令}{\begin{原环境命令注解}}
\newcommand{自定义环境结束控制命令}{\end{原环境命令注解}}
```

例如将配对的环境控制命令的化简程序

```
\newcommand{\be}{\begin{equation}}
\newcommand{\ee}{\end{equation}}
```

插入在全局说明区域内, 则可将常用的独立编号数学方程命令

```
\begin{equation}
<数学方程式>
\end{equation}
```

变换成完全等价的简单形式:

```
\be <数学方程式> \ee
```

### 10.2.2 自行定义排版环境控制命令

自行定义一个排版环境控制命令并不容易, 需要考虑许多在排版过程中可能出现的各种因素。由于 $\text{\LaTeX}$ 系统已经设置了一些功能非常齐全的排版环境, 一般情况下, 用户只须借助于已有的环境命令, 附加一些排版功能来定义一个新的排版环境控制命令。所以, 与其说自行定义一个排版环境控制命令, 倒不如说是环境名变换更确切些。 $\text{\LaTeX}$ 语言中将已有的排版环境控制命令中的环境注解参量作更换, 并且附加一些排版注解的命令格式为:

```
\newenvironment{自定义环境注解}
{\begin{原环境注解} 附注}{\end{原环境注解}}
```

说明:

① 自己新定义的环境命令的使用方法与第四章中所介绍的环境命令的使用方法类似, 其标准格式为

```
\begin{自定义环境注解}
<正文部分>
\end{自定义环境注解}
```

② 自定义的环境注解名不能与系统已经存在的环境注解同名, 否则排版时会出错。

③ 上述命令中的附注 是泛指紧接自定义的环境起始命令以后的其它排版控制命令或源程序。例如程序:

```

\newenvironment{quorem}{\begin{quotation}\small\em}{\end{quotation}}
.....
\begin{quorem}
Here is an example of the quotation environment which
causes text to be set somewhat narrower than the normal
and switches from the normal type to the emphatic style
with smaller type size (nine points in ten-point paper).

\end{quorem}

```

可由常规的引文环境(quotation)再定义一个相对正文基准字小一号,并且选择强调字体的新引文环境(quorem),其输出拷贝为:

*Here is an example of the quotation environment which causes text to be set somewhat narrower than the normal and switches from the normal type to the emphatic style with smaller type size (nine points in ten-point paper).*

### 10.2.3 含变量排版环境的自定义方法

L<sup>A</sup>T<sub>E</sub>X 系统的绝大部分排版环境控制命令是不含参变量的。利用现有的排版环境,可以自行设计带参量的排版环境控制命令。类似在L<sup>A</sup>T<sub>E</sub>X 排版子过程控制命令中嵌套 $n$ 个虚拟参变量的设置方法(见第10.1.2),在特殊排版需要时,亦可将自定义的排版环境命令作些扩充。假若用户要自行定义一个带 $n$ 个参量的L<sup>A</sup>T<sub>E</sub>X 排版环境控制命令,可按如下格式来编程:

```

\newenvironment{自定义环境注解}[ $n$ ]
{\begin{原环境注解}带 $n$ 参量的附注}{\end{原环境注解}}

```

说明:

- ① 其中带 $n$ 参量的附注是泛指形如

... #1 ... #2 ..... # $n$  ...

的带 $n$ 个虚拟参量的排版控制命令或其它源程序,这里“# $i$ ”表示嵌在排版附注中的第 $i$ 个虚拟参变量,而“...”则表示其它而定的辅助排版说明语句。② 方括号内的参数“ $n$ ”是用来确定自定义排版环境控制命令中所含参数的个数。③ 自己新定义的带参量的环境命令的使用方法与第四章中所介绍的环境命令的使用方法略有不同。用户在输入新环境起始命令时,还须输入 $n$ 参变设。具体的调用格式为

```

\begin{自定义环境注解}(参量1)(参量2){.....参量 $n$ }
<正文部分>
\end{自定义环境注解}

```

④ 自定义的带参数的排版环境命令不能与系统已经存在的环境命令同名, 否则排版时会出错。

⑤ 辅助排版说明语句中不能出现任何带有“\end”形式的控制命令, 否则定义失效。

例如, 我们可以将引文环境(quote)改造成带三参量(论文题目、作者名和单位)的摘要排版环境(abst)

```
\newenvironment{abst}[3]{\begin{quote}{\large\bf #1}\
{\rm #2}\({\footnotesize \em #3\/})}\
{\bf Abstract:} \sl}{\end{quote}}
```

调用此环境的排版实例如下:

```
Here is an example of using a user-defined abstract
environment:
\begin{abst}
{Reflection of soft x-rays from a finite quasi-periodic
superlattice}{Nian-hua Liu, Wei-guo Feng and Xiang Wu}
{Department of Physics, Tongji University, Shanghai
200092, P R China}
The normal incident reflectivity of soft x-rays from a
finite quasi-periodic superlattice (QPSL), in which all
cells have the same optical thickness period but the
optical thickness of the high absorbing layer increases
gradually from the surface to the bottom of the QPSL, has
been investigated by using the transfer-matrix method
\ldots\ldots\ldots
\end{abst}
```

输出结果为:

Here is an example of using a user-defined abstract environment:

**Reflection of soft x-rays from a finite quasi-periodic superlattice**

Nian-hua Liu, Wei-guo Feng and Xiang Wu

(Department of Physics, Tongji University, Shanghai 200092, P R China)

**Abstract:** The normal incident reflectivity of soft x-rays from a finite quasi-periodic superlattice (QPSL), in which all cells have the same optical thickness period but the optical thickness of the high absorbing layer increases gradually from the surface to the bottom of the QPSL, has been investigated by using the transfer-matrix method .....

### 10.2.4 重新定义已经设置的排版环境

L<sup>A</sup>T<sub>E</sub>X 系统为用户定义了一些标准的排版环境。如果需要增加一些排版功能, 用户还可以由现有的排版环境再自行设计一些新的排版环境。但是自定义的排版环境注解名绝对不能与系统保留的或已经自行定义的排版环境同名。如果出现同名, 则会产生排版错误(系统会给用户出错报警)。这时, 比较安全的方法是另外选取一个环境注解名。如果用户需要将现有的 L<sup>A</sup>T<sub>E</sub>X 排版环境的功能作某些修改(不打算更改环境注解名), 就要对原有的环境重新定义。L<sup>A</sup>T<sub>E</sub>X 系统中提供的对现有的排版环境重新定义的控制命令为:

```
\renewenvironment
```

其与自定义环境命令“\newenvironment”的使用方法完全类似, 只是被定义的对象不同而已。如果用户没有弄清楚自己究竟在干什么, 千万不要轻易修改 L<sup>A</sup>T<sub>E</sub>X 系统设置的标准排版环境。

最后要提醒用户注意的是自定义的排版环境命令可以与现有的 L<sup>A</sup>T<sub>E</sub>X 排版环境或已经定义好的其它排版环境控制命令相互嵌套使用, 但是在排版环境自定义语句中绝不能嵌套使用以下控制命令中的任何一个:

```
\newcommand
\renewcommand
\newenvironment
\renewenvironment
```

换句话说, 上述四个命令不能相互嵌套使用。

## 10.3 人机对话

所谓人机对话, 首先是指 L<sup>A</sup>T<sub>E</sub>X 在执行排版过程中, 系统在终端屏幕上显示一些提醒样式的文字信息; 其次是指如果系统在排版过程中版面上需要某些可变动的文字信息(例如姓名、地址和电话号码等), 用户可以通过键盘及时输入。L<sup>A</sup>T<sub>E</sub>X 在执行排版时实现人机对话功能的控制命令主要有两种: 终端显示命令和终端输入命令。

### 10.3.1 终端显示命令

排版文字信息在终端显示的控制命令格式为

```
\typeout{西文文字信息}
```

如果系统执行排版时在源程序中读到了上述格式的控制命令, L<sup>A</sup>T<sub>E</sub>X 会立即向终端屏幕输出在命令花括号中的文字信息。例如某一 L<sup>A</sup>T<sub>E</sub>X 源文件(sample.tex) 中插入了以下命令



```

.....
\typeout{Don't forget to copy LaTeX file to the Disk!}
.....

```

则在执行排版时, 终端屏幕会显示如图10.1所示的几行信息:

```

.....
Don't forget to copy LaTeX file to the Disk!
.....
[1] (sample.aux)
Output written on sample.dvi (1 page, 2340 bytes).

```

图 10.1 提示信息终端显示示意图

### 10.3.2 终端输入命令

#### 一、终端输入文字信息

在执行排版过程中, 通过键盘在终端输入有关文字信息的控制命令格式为

```
\typein{终端提示文字信息}
```

如果 $\text{\LaTeX}$ 在源程序中读到了上述控制命令, 则会先向屏幕输出花括号中的终端提示文字信息, 然后在下面显示一串待命字符:

```
\@typein=
```

当用户输入一行西文文字并按下回车键时, 系统便立即将输入的文字编排在当前位置, 然后继续执行后面源文件的排版工作。例如在 $\text{\LaTeX}$ 源文件(sample.tex)中插入了以下有关输入用户姓名(name)和工作单位(institute)的控制命令:

```

.....
This is a first example of using command \verb+\typein+ \\
.....
My name is \bf \typein{Please input your name here:},
\rm and I am working in the \bf
\typein{Please input your institute here:}. \rm
.....

```

在执行排版时, 如果用户根据提示先后输入了自己名字和单位的西文字符串, 并分别按下了回车键(如图10.2所示, 图中标有下划线的字符即为键盘输入的信息), 那么, 系统便会把用户的名字和单位分别编排在命令所处的位置, 这样你便能获得以下拷贝:

```

.....
Please input your name here:

\typein=FENG Wei-guo ✓
Please input your institute here:

\typein=Tongji University ✓

```

图 10.2 文字信息终端输入示意图

```

.....
This is a first example of using command \typein
.....
My name is FENG Wei-guo, and I am working in Tongji University.
.....

```

## 二、为自定义的控制命令赋字符串

在执行排版过程中，如果通过终端键盘输入的文字信息在以后的排版过程中会多次被引用，则可定义一个可赋字符串的新的打印控制命令。以后排版需要时，可直接调用该命令。为自定义的控制命令赋字符串的命令格式如下：

```
\typein[自定义字符串命令]{终端提示文字信息}
```

当L<sup>A</sup>T<sub>E</sub>X 读到此命令时，就会为用户定义一个新的可输出字符串的打印控制命令，例如前面介绍的排版程序可改为

```

.....
\typein[\name]{Please input your name here:}
\typein[\institute]{Please input your institute here:}
This is a second example of using command \verb+\typein+ \\
.....
My name is {\bf \name},
and I am working in {\sf \institute}. I believe you
are also come from the {\sf \institute}.
.....

```

在执行排版时，可按照图10.3 所示的输入方式(图中标有下划线的字符) 将自己的名字和单位的西文字符串通过键盘分别键入。那么，系统便会把用户的名字和单位分别编排在命令所处的位置。排版完毕后，你便能获得以下拷贝：

```

.....
Please input your name here:

\name=Mo Hui-lin ✓
Please input your institute here:

\institute=Tongji University ✓

```

图 10.3 为自定义控制命令赋字符串的屏幕显示示意图

```

.....
This is a second example of using command \typein
.....
My name is Mo Hui-lin, and I am working in the Tongji University. I believe that
you are also come from the Tongji University.
.....

```

## 10.4 调入另外一个 $\text{\LaTeX}$ 源文件

从本书开篇至此页，所列举的排版程序基本上都假定是属于一个完整 $\text{\LaTeX}$ 源文件中的某一部分。所谓完整，是指假文件中至少要有文件格式命令，文稿起始命令和文稿结束命令。否则， $\text{\LaTeX}$ 将无法执行排版。对于篇幅不太长的科技论文或报告，如果将含有排版控制命令的全部文稿输入在一个文件内，不会在执行排版或编辑过程中给用户带来多大麻烦。但是，对于页数较多的书刊类文件，编排和运行一个信息量极大的 $\text{\LaTeX}$ 假文件，显然很不方便。假定用户只想对文件中的某一页作些修改(例如含有100多页内容文件中的第80页)，那么在排版时，不仅每次寻找和编辑该页内容非常困难，而且在进行调试(执行排版)时还要耗费许多不必要的机时，这样白白浪费了用户的宝贵时间。

解决上述问题的最简单方法是化假为零，即将正文文稿分成几个独立的可执行源文件(在每个文件中分别添加文件格式命令，文稿起始命令和文稿结束命令)，待各自调试完毕后再合并在一起。但是如果遇到需要交叉引用的情形，在分开调试时，各个独立的文件之间就无法沟通排版信息。 $\text{\LaTeX}$ 系统在处理化整为零问题时，采用了一假程序语言中规范的在主文件(称之为根文件)中调入其它相关文件(叫作子文件)的程序设计方法。这样，既可以方便地编辑假文件，又能有选择地调入有关子文件进行排版调试，由于系统在磁盘内存贮了排版过程中各个独立子文件的排版信息，所以即使在分散地编排文件时，也能较好地实现交叉引用。

### 10.4.1 无选择方式调入 L<sup>A</sup>T<sub>E</sub>X 子文件

假定用户已经有了一个可执行的根文件(排版主程序), 现在希望在对根文件执行排版时, 无条件地调入另外一个 L<sup>A</sup>T<sub>E</sub>X 子文件(排版子程序), 并使之与根文件合并在一起排版, 那么可以使用如下调入了文件控制命令:

```
\input{LATEX 子文件名}
```

说明:

① L<sup>A</sup>T<sub>E</sub>X 子文件名 是指需要调入的另外一个 L<sup>A</sup>T<sub>E</sub>X 源文件引用名, 这个文件必须是已经事先建立好, 其扩展名必须是“TEX”, 并且已存放在当前目录(例如在 PCTEX 子目录)中。扩展名不要注明, 因为 L<sup>A</sup>T<sub>E</sub>X 读到调入了文件语句时, 会自动在当前目录中搜寻所要调入的扩展名为“TEX”的排版源文件。

② 除了可执行的根文件之外, 所有要调入的子文件一般都不必添加文件格式命令, 文稿起始命令和文稿结束命令。因为“\input”命令生效时, 这些离散的源文件就被直接插入在命令出现的位置。当然, 如果根文件只有一行 \input 命令, 那么必要的文件格式等排版命令可以出现在后续的子文件中。

③ 在调入的子文件中可继续使用“\input”命令, 从而可再插入其它有关的下一级子文件。因此, 用户可以根据实际需要由根文件出发, 按树根分叉原理逐级嵌套使用“\input”命令。

④ “\input”命令可以插入在 L<sup>A</sup>T<sub>E</sub>X 源文件中的任何位置(包括全局说明区域)。

⑤ 如果“\input”命令中指定的子文件不存在, 系统会向用户报错:

```
! I can't find file '...'.

```

(文件不存在)。此后, 屏幕上显示期待输入另外一个子文件名的信息:

```
Please type another input file name:
```

你可通过键盘, 再输入其它确定存在的子文件名。按回车键后, 系统会继续执行排版工作。

作为本节内容的排版实例, 我们介绍利用 L<sup>A</sup>T<sub>E</sub>X 人机对话功能来编制一个非常有用的调试小块源文件的 L<sup>A</sup>T<sub>E</sub>X 排版主程序(假定排版子程序中既没有文件格式命令, 也没有文稿起始命令和结束命令, 如本书中列举的大量源程序实例):

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% FILENAME: test.tex %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\documentstyle{article}
\textwidth=16cm
\textheight=23cm
\oddsidemargin=0pt\evensidemargin=0pt\topmargin=0pt
\parindent=0pt
```

```

\pagestyle{headings}
%% You can add more preamble command here .....
\begin{document}
\typeout{Main program for testing seperate TEX file}
\typein[\texfile]{Enter name of TEX file:}
\input{\texfile}
\end{document}

```

对此程序运行 $\text{\LaTeX}$ , 用户就可以键入需要调试的子程序文件名, 按回车后即可进行排版处理。

#### 10.4.2 有选择调入 $\text{\LaTeX}$ 子文件

“ $\text{\input}$ ”命令虽然解决了化整为零的问题, 但是在插入若干条“ $\text{\input}$ ”命令之后, 对嵌套组成的大型文件执行排版工作仍然要耗费大量时间。要想有选择调入 $\text{\LaTeX}$ 子文件, 只能通过在各个源文件中删改“ $\text{\input}$ ”命令来进行有选择地调试工作。这对用户来说, 仍然是不太方便的。为此,  $\text{\LaTeX}$ 系统又提供了一个可以有选择地调入 $\text{\LaTeX}$ 子文件的排版控制命令:

$\text{\include{ $\text{\LaTeX}$ 子文件名}}$

说明:

① 如果没有在全局说明区域内对“ $\text{\include}$ ”命令作选择方式的任何说明, 其调入 $\text{\LaTeX}$ 子文件的功能与命令“ $\text{\input}$ ”相类似。只是命令生效时会另面排版。

② 要想使调入 $\text{\LaTeX}$ 子文件选择功能生效, 则必须首先在全局说明区域内输入子文件选择命令。假定用户在原来排版程序中所要调入的子文件共有 $n$ 个:

子文件名1, 子文件名2, ..., 子文件名 $n$

如果现在为了调试方便(节约时间), 只希望调试其中的几个, 那么用户不必在源程序抹去暂时不需要的“ $\text{\include}$ ”命令, 而是在全局说明区域内输入如下命令:

$\text{\includeonly{ $\text{\LaTeX}$ 子文件名 $i$ , 子文件名 $j$ , ...}}$

这样, 系统在执行排版时, 选择功能即可生效。在命令“ $\text{\includeonly}$ ”中注明的子文件名将被分别调入, 而其它的只是用“ $\text{\include}$ ”命令注明的子文件将被忽略。

③ 虽然在命令“ $\text{\includeonly}$ ”中没有注明的子文件名不会在执行排版过程中调入, 但是只要这些源文件曾经被调试过(在磁盘中留下有关的排版信息, 例如相应的AUX文件, 目录(TOC)文件等), 那么即使在调试其它部分的排版子程序(子文件)时, 仍旧可以实现交叉引用。这些未被调入的文件就好象是实际排版源程序中一部分, 在执行排版过程中仍然会对其它部分的文稿排版样式产生作用(例如页码、章节号数字可保持延续性)。

④ 命令“\include”在使用时有一定局限性，因为系统读到此命令时会调入的子文件内容立即无条件地另起一页排版。所以一般在按章次独立编排源文件时使用比较方便。

⑤ 为了保证最后输出的版面页码准确无误，用户在确定分成几个部分的源程序整体不再修改后，应将所有子文件一起调入(在根文件中将选择命令删除)再令L<sup>A</sup>T<sub>E</sub>X连续执行排版两次。如果你不喜欢命令“\include”导致另面排版的方式，则在确定最后版本时，可将所有命令改成“\input”，再执行排版。

## 10.5 如何改变系统的计数方式

### 10.5.1 计数器和计数方式

L<sup>A</sup>T<sub>E</sub>X 智能排版系统的优点之一是能为用户自动编排各种带序号的版面内容，如页码、脚注、章节等。对任一带序号的排版项目，系统都设计了一个与之对应的计数器(counter)。表10.1列出了常用的L<sup>A</sup>T<sub>E</sub>X排版系统中可编号的项目以及相应的计数器名(标识用参量)。

表 10.1 L<sup>A</sup>T<sub>E</sub>X 排版计数器标识名表

项目	计数器标识名	项目	计数器标识名	项目	计数器标识名
篇	part	小段	subparagraph	方程	equation
章	chapter	页	page	图	figure
节	section	一级条目	enumi	表	table
款	subsection	二级条目	enumii	小页脚注	mpfootnote
项	subsubsection	三级条目	enumiii	脚注	footnote
段	paragraph	四级条目	enumiv		

各项目的序号编排方式主要有两种：

(1) 单一正整数 仅由一个计数器确定的整数值(1, 2, 3, ...), 如篇、脚注、页等使用的序号。

(2) 复合正整数 由两个或三个计数器共同确定的用小数点连接的复合整数值。例如节号(1.1, 1.2, 1.3, ...)中的第一个整数1来之章(chapter)次序号，而第二位数才是节(section)计数器给出的数值。

### 10.5.2 为计数器赋值

任何一个计数器在初始状态时，系统会先自动将其赋值为0；在执行排版的过程中，每被调用一次，系统便自动为其累加数值1。如果需要在对源程序执行排版的某一过程中改变某个项目计数器的数值，可以在源程序再插入一个为计数器重新赋值的控

制命令:

`\setcounter{计数器标识名}{n}`

说明:

- ① 计数器标识名 可以是表10.1 中的任意一个。
- ② 参数 $n$  是用户希望设置的特定正整数。在赋值控制命令以后, 相应的计数器在被第一次调用时输出的数值将是 $n + 1$ 。

如果用户不能确定给计数器所赋数值的大小, 而只是要在计数器原有的数值上增加或减少某个量值, 那么还可以使用另外一个为计数器添加数值的控制命令:

`\addtocounter{计数器标识名}{ $\Delta n$ }`

说明:

- ① 这里参数 $\Delta n$  是为计数器添加的增量, 可以是正或负整数。
- ② 假定计数器原有的数值为 $n$ , 则在添加数值命令以后第一次被调用时输出的数值为 $n + \Delta n + 1$ 。

### 10.5.3 计数器数字打印命令

如果要想打印出计数器处于某一状态下的数字(如页码, 章节号等), 可以使用如下形式的计数器数字打印命令:

`\the 西文计数器标识名`

这里用“`\the`”打头引出的西文计数器标识名可以选取表10.1 中所列举的任意一个。例如打印处于当前排版状态下的页码和章、节序号可分别由以下三个控制命令来实现:

`\thepage`                      `\thechapter`                      `\thesection`

### 10.5.4 改变序码的字体形式

尽管计数器内的所含的数值是正整数, 但在执行排版后, 打印输出的数字字体还可以另外再选择。LaTeX 系统提供的数字字体共有五种, 具体控制计数器数字字体的命令以及实际打印的数果可参见表10.2。

注意其中的采用英文大小写字体的数字不能大于26, 此外脚注专用序码的实际序号值也不能大于9(有些特殊字符在调用时须设置数学排版状态)。在需要改变计数器数字字体时, 可以采用命令

`\renewcommand`

来重新定义数字打印命令。

表 10.2 序号数字字体控制命令表

数字字体	数字字体控制命令	实际计数和打印样式
阿拉伯数字	<code>\arabic{计数器标识名}</code>	1, 2, 3, 4, 5, 6, 7, .....
罗马小写数字	<code>\roman{计数器标识名}</code>	i, ii, iii, iv, v, vi, vii, .....
英文小写字母	<code>\alph{计数器标识名}</code>	a, b, c, d, e, f, g, .....
罗马大写数字	<code>\Roman{计数器标识名}</code>	I, II, III, IV, V, VI, VII, .....
英文大写字母	<code>\Alph{计数器标识名}</code>	A, B, C, D, E, F, G, .....
脚注专用序码	<code>\fnsymbol{计数器标识名}</code>	*, †, ‡, §, ¶,   , **, ††, ‡‡

例如, 以下程序可将脚注序号的打印字体由阿拉伯数字改为脚注自身专用序码(见本页脚注序号)<sup>1</sup>:

```
\renewcommand{\thefootnote}{\fnsymbol{footnote}}
This is a real example of using footnote symbol.
Note that the present number of footnote counter is
'$\thefootnote$', and the \verb|\setcounter| command sets
the value of the footnote counter from '$\thefootnote$' to
\setcounter{footnote}{6} '$\thefootnote$'.
\footnote{The value of the counter must be less than 10.}
```

输出时, 你除了可得到以下拷贝外, 还能看到本页底部的第二个脚注。注意其序码由于在源程序中改变了数值, 从而由“1”转变为“\*\*”。

This is a real example of using footnote symbol. Note that the present number of footnote counter is “\*”, and the `\setcounter` command sets the value of the footnote counter from “\*” to “||”. \*\*

又如在美国科技刊物中论文节次(section)的序号普遍采用大写罗马字体。需要这种版式的用户可以在自己源程序的全局说明区域内输入如下命令:

```
\renewcommand{\thesection}{\Roman{section}}
```

在节次(section)以下层次, 如款(subsection)的序号亦可以自行设计。假定用户需要将款的序码打印成由节次和款次(用大写英文字母)复合组成的形式, 那么, 紧接着上面一程序, 可再输入一条命令:

```
\renewcommand{\thesubsection}{\thesection--\Alph{subsection}}
```

<sup>1</sup>如果整篇文章或整本书的脚注序码均采用某种数字字体时, 可将脚注的重新定义命令放在源文件的全局说明区域内。

\*\*The value of the counter must be less than 10



输出时对于第二节中的节款序号将为

```

:
II .....
:
II-A .....
:
II-B .....
:

```

## 10.6 自定义排版计数器

使用排版计数器的自定义控制命令，用户可以产生一个新的计数器，命令格式为：

```
\newcounter{自定义计数器标识名}[已有的计数器标识名]
```

其中自定义计数器标识名 参量由合法的英文字母定义，而方括号中的参量已有的计数器标识名 是标识已由系统定义的计数器名(注意此命令只能在全局说明区域内使用)。如果省略后面的参量，则自定义的计数器初值自动为零；如果由已有的计数器来定义新的计数器，则在任何时候，都可使用以下命令将其初值重新赋为零：

```
\stepcounter      \refstepcounter
```

当需要在正文中打印计数器的数值(阿拉伯数字形式)时，可使用命令

```
\the自定义计数器标识名
```

如果需要改变数字字体，则可使用第10.5.4 小节介绍的计数器数字字体转换命令。

## 10.7 定理排版环境命令

数学类西文文章中常出现一些定理、引理、推论等相同类型的版面。这种版面有点象描述型列表条目，其特点是总有一个需要强调的显著标题，如“**Theorem**”(定理)、“**Conjecture**”(假设)等；定理之间的区分主要靠标题后面的序号来确定；后面的正文须用强调字体打印出来。在非数学类西文文章中也会出现一些类似结构的版面，如定律(Laws)、原理(Principles)和规则(Rules)等。为了使用户非常方便编排这样的规则版面， $\text{\LaTeX}$  提供了一种定理类型的特殊排版环境。

定理排版环境命令由两部分组成。首先是要对定理排版命令中的环境注解名进行定义，命令格式如下：

```
\newtheorem{定理环境注解名}{定理标题}[章节序号参量]
```

其中章节序号参量是对定理编号方式作注解, 当取参量“chapter”时定理编号层次限于定理所处的章以内, 而取参量“section”时则定理编号层次限于定理所处的节以内。章节序号参量可以省略, 省略时编号只是采用与章节序号无关的独立正整数。定理环境注解名的定义控制命令最好是放在全局排版说明区域内, 但也可放在源文件的正文之中的任何位置。环境注解名可按常规取任意的合法字符串(不要使用配对的方括号)。有了预先定义好的环境注解名, 在文中需要编排该定理时就可使用如下方式的排版环境命令:

```
\begin{定理环境注解名}
<定理正文>
\end{定理环境注解名}
```

例如:

```
\newtheorem{Bohr}{Assumption}[chapter]
.....
\begin{Bohr}
The electrons move in circular orbits about the center
of the atom.
\end{Bohr}
\begin{Bohr}
The only allowed orbits are such that the angular
momentum of the atom about its center of mass is an
integral multiple of  $h/2\pi$ 
\end{Bohr}
\begin{Bohr}
Radiation occurs only when an electron ‘‘jumps’’ from one
of the allowed orbits to one of lower energy. The difference
in energy  $\Delta E$  is the radiated as a photon of
frequency  $\nu = \Delta E/h$ , in accordance with the
Einstein condition.
\end{Bohr}
```

可得:

.....

*Assumption 10.1 The electrons move in circular orbits about the center of the atom.*

*Assumption 10.2 The only allowed orbits are such that the angular momentum of the atom about its center of mass is an integral multiple of  $h/2\pi$*

**Assumption 10.3** *Radiation occurs only when an electron “jumps” from one of the allowed orbits to one of lower energy. The difference in energy  $\Delta E$  is the radiated as a photon of frequency  $\nu = \Delta E/h$ , in accordance with the Einstein condition.*

如果用户在已经定义并编排好的定理类型的文件后面按顺序再续接另外一个变换了标题的定理类文件(即把两种标题按一个编码序列标号), 那么用户还可使用如下格式的定义控制命令:

```
\newtheorem{定理环境注解名1}{定理标题1}
\newtheorem{定理环境注解名2}[定理环境注解名1]{定理标题2}
```

例如在下面程序中:

```
\newtheorem{Rontgen1}{Property}
\newtheorem{Rontgen2}[Rontgen1]{Characteristic}
Röntgen, after a six-week series of experiments,
published an article listing many fundamental properties of
the rays. He found:
\begin{Rontgen1}
The rays are produced when the cathode rays strike a solid
surface, such as the anode of the cathod-ray tube.
\end{Rontgen1}
\begin{Rontgen2}
The rays are not deflected by magnetic fields (and hence
they must be unchanged.
\end{Rontgen2}
.....
```

将不同的标题并入了统一的编码序列, 在输出时, 可以获得如下结果:

Röntgen, after a six-week series of experiments, published an article listing many fundamental properties of the rays. He found:

**Property 1** *The rays are produced when the cathode rays strike a solid surface, such as the anode of the cathod-ray tube.*

**Characteristic 2** *The rays are not deflected by magnetic fields (and hence they must be unchanged.*

.....

173

## 10.8 列表环境(list)的构造

“list”是泛指以规则的条状形式排列成的表。实际上L<sup>A</sup>T<sub>E</sub>X定义的许多环境，如quote、quotation、verse、itemize、enumerate、description、thebibliography、center等等，都是以“list”列表环境形式构造而成的。“list”的特点是条状正文版面宽度一般比当前版心的版面要短些，两边均留出较宽白边；此外，左上角总有标记此条状正文(条目)的标签，当然有时标签可以省略；再则，条目正文的自然段起始时可以缩排，也可以顶格排。

在第4.8节中介绍了L<sup>A</sup>T<sub>E</sub>X排版系统定义的专用条目列表环境命令，其特点是版面形式固定不变。用户在使用这些环境命令时，只须在条目列表环境中填充每一条项目的具体内容就可以了。至于条目正文与版心边界的距离、条目之间的空距、条目标签与条目正文的间距等等，用户完全用不着去考虑。但是，假如用户希望自己定义一个类似条目列表环境的控制命令，则可使用允许用户自行构造条目列表环境的带二参量的list环境控制命令：

```
\begin{list}
{条目标签形式}
{条目版式控制命令}
\item_1 <条目1>
\item_2 <条目2>
.....
\item_n <条目n>
\end{list}
```

其中参量条目标签形式定义了一个LR排版盒子，盒子内既可以是字符串，也可以是自行定义的计数器，或者是两者的组合。当引出<条目n>的控制命令“\item”后面未标识标签形式时系统会采用参量条目标签形式。第二个参量条目版式控制命令可根据排版具体需要使用一些版式控制命令，各条命令在版面上的控制功能如图10.4所示。

- \topsep 第一条项目的正文顶部与前面文字之间的纵向空距；最后一条项目的正文底部与后续文字之间的纵向空距。
- \partopsep 假定源程序的第一条项目的正文顶部与前面的文字之间有一划分自然段的空行，那么此命令的作用是另外再插入一定空距。
- \itemsep (除了由“\parsep”命令插入的空距外)在条目之间增加的纵向空距。
- \parsep 当某个条目分成若干自然段时，此命令在自然段间插入一定的空白空距。
- \leftmargin 条目版面的左边界与版心左边界的距离。如果需要赋值，则必须是带长度单位的正实数。

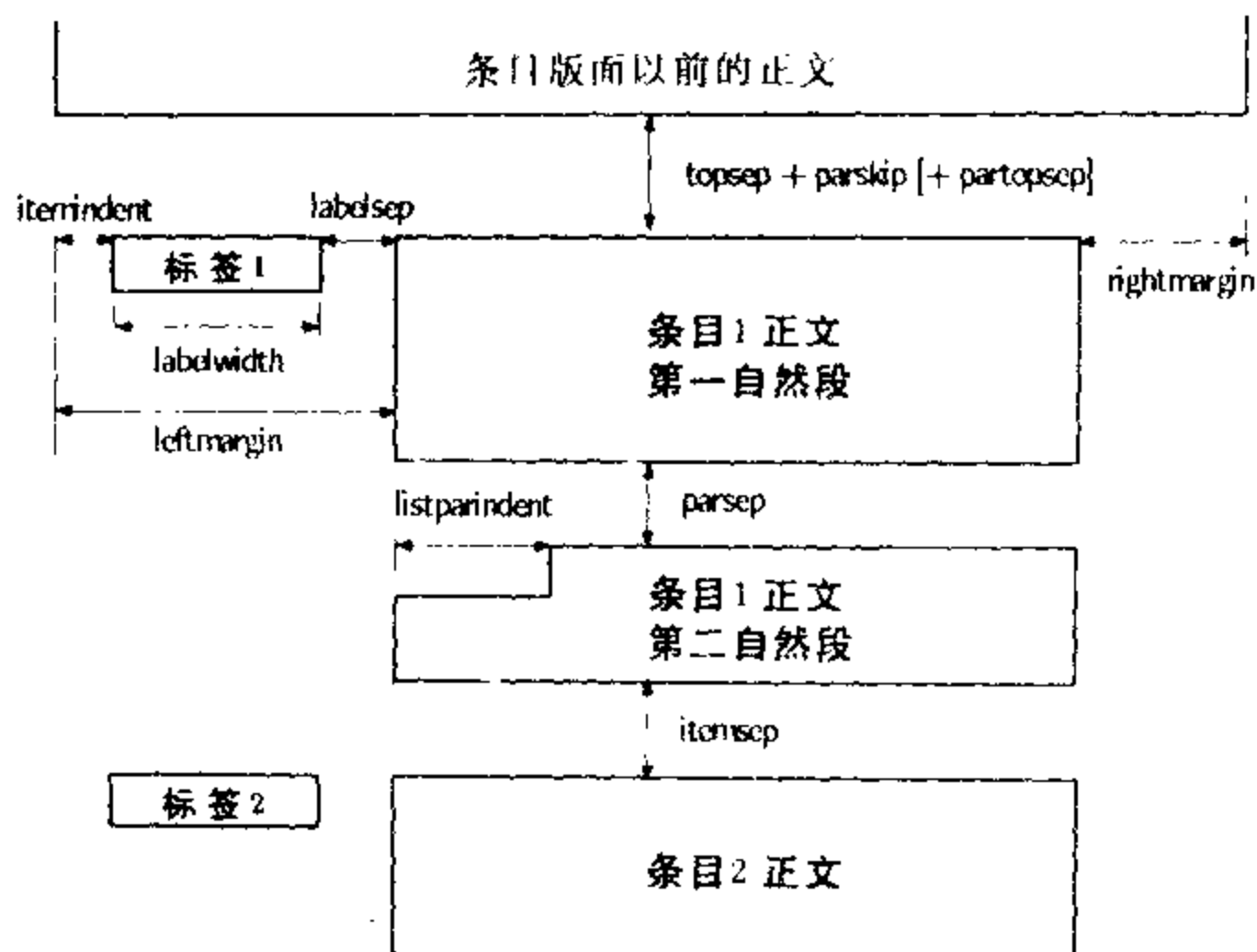


图 10.4 条目列表环境(list) 版面控制示意图

- `\rightmargin` 条目版面的右边界与版心右边界的距离。如果需要赋值，则必须是带长度单位的正实数。
- `\listparindent` 每个自然段第一行缩排时从条目版面的左边界缩进的总长度(可取负值)。
- `\itemindent` 条目标签距离版心的左边界空出的长度(可取负值)。
- `\labelsep` 条目标签与条目正文之间的距离。
- `\labelwidth` 标签横向总长度。
- `\usecounter` 使用自定义计数器。

以上凡是涉及尺度的控制命令可在全局说明区域，或是在当前环境内进行更改，如果未予赋值，则系统自动取省略状态下的数值。下面是一个使用自定义计数器的list环境排版实例：

```
\newcounter{ncname}
Here is a new example of the list environment{\CC {\char35}.
}\begin{list}
{C--\Roman{ncname}}
{\usecounter{ncname}\setlength{\rightmargin}{\leftmargin}}
\item This is a first item of the list. Observe how the left and
```

```

right margins are indented by the same amount.
\item This is the second item of the list. Note that the labels
are automatically numbered with Roman type.
\item This is the last item of the list.
\end{list}

```

输出结果:

Here is a new example of the list environment.

C-I This is a first item of the list. Observe how the left and right margins are indented by the same amount.

C-II This is the second item of the list. See the label are automatically numbered with Roman type.

C-III This is the last item of the list.

## 10.9 如何使用新的字体

在本书第3.2节中,我们介绍了L<sup>A</sup>T<sub>E</sub>X系统的几种标准西文字体。对于一般科技论文的排版,这些字体已经足够用了。但是对于一些特殊的文艺书刊,有时还需要有更丰富的字体来修饰版面。现在比较流行的个人用L<sup>A</sup>T<sub>E</sub>X排版系统(例如PCT<sub>E</sub>X软件)一般都配置了较多的字体供用户选择。如果用户手头有现成的排版系统说明书,那么,就可以查阅到可使用的新字体详表和打印样式。如果没有用户手册,那么可以在磁盘的“TEXTFMS”(PCT<sub>E</sub>X软件)或“TFM”(E<sub>M</sub>T<sub>E</sub>X软件)子目录中查找需要使用的字体。例如对装有2.09版本L<sup>A</sup>T<sub>E</sub>X排版系统的PCT<sub>E</sub>X软件,其子目录“TEXTFMS”中的排版字库文件(扩展名为“TFM”)如下所示:

Directory of C:\PCTEX\TEXTFMS

..	..	AMB10	TFM	AMBC12	TFM	AMBI10	TFM
AMBSY10	TFM	AMBI10	TFM	AMBI5	TFM	AMBI7	TFM
AMBI5	TFM	AMBI9	TFM	AMBI5L10	TFM	AMCSC10	TFM
AMEX10	TFM	AMINCH	TFM	AMITT10	TFM	AMNI10	TFM
AMMI5	TFM	AMMI7	TFM	AMMI8	TFM	AMMI9	TFM
AMR10	TFM	AMR5	TFM	AMR6	TFM	AMR7	TFM
AMR9	TFM	AMRC10	TFM	AMRTEX10	TFM	AMSL10	TFM
AMSL9	TFM	AMSLIT10	TFM	AMSS10	TFM	AMSSB110	TFM
AMSSMC10	TFM	AMSSMC40	TFM	AMSSQ8	TFM	AMSSQ18	TFM
AMSY5	TFM	AMSY6	TFM	AMSY7	TFM	AMSY8	TFM
AMTEX10	TFM	AMTI10	TFM	AMTI7	TFM	AMTI8	TFM
AMTI10	TFM	AMTI8	TFM	AMTI9	TFM	AMU10	TFM
CIRCLEW1	TFM	DUMMY	TFM	LASY10	TFM	LASY5	TFM
LASY7	TFM	LASY8	TFM	LASY9	TFM	LINE10	TFM
MANFNT	TFM					LINEW10	TFM

差不多所有的西文字体文件都冠以“am”为前缀(在EMTeX中以“cm”为前缀),而L<sup>A</sup>T<sub>E</sub>X特殊符号的字库文件为“lasyn”。其中常用的各种字号的字体所对应的库文件如表10.3所示(其中n标识5-10之间的整数,表示点(pt)数)。

表 10.3 常用排版字库文件引用名表

字体	TFM 文件引用名	字体	TFM 文件引用名
罗马字体	amrn	行文意式斜体	amtin
黑体	ambxn	数学意式斜体	ammin
等线体	amssn	倾斜体	amsln
打印机字体	amttn	小体大写字母	amscsn
L <sup>A</sup> T <sub>E</sub> X 特殊符号	lasyn		

除了表10.3中所列举的常用字库文件之外,在子目录中还会有一些其它字体的文件。这些就为用户提供了调用新字库的可能。例如文件“AMBXSL10.TFM”的存在提供了用户使用倾斜黑体的机会。附录A中列举了PCT<sub>E</sub>X和EMTeX系统中可以使用的各种英文字体及对应的字库文件名。

常用的各种字号的字体在启动L<sup>A</sup>T<sub>E</sub>X之前就被装入系统,用户可以在源文件中直接插入第3.2节中介绍的字体转换命令就能用各种字体来编排文件。但是对于并不常用的字体,用户需要时必须先用新字体命令将其装入系统,然后才能在行文排版中调用。调入新字体的控制命令格式如下:

```
\newfont{新字体转换命令}{TFM文件名}
```

例如,调入倾斜黑体的源程序为:

```
\newfont{\bfsl}{ambxsl10}
```

说明:

① 自定义的新字体转换命令必须是以反斜杠“\”为前缀的英文字符串。在此命令后面的行文或LR排版状态中,用户可用新字体转换命令来切换字体。例如:

```
\newfont{\bfsl}{ambxsl10}
.....
Here you will see a type of new font named as\
'\bfsl Boldface Extended Slanted Roman'.
```

输出时就会得到倾斜黑体字的排版结果:

```
.....
Here you will see a type of new font named as
Boldface Extended Slanted Roman.
```

② 在数学状态中, 自定义的新字体命令必须嵌套在LR 盒子命令“\mbox”中使用。

③ 由于调入系统的新字体转换命令并没有在L<sup>A</sup>T<sub>E</sub>X 系统事先给予定义, 所以, 不能简单地用字号转换命令来改变新字体的大小, 如果需要改变大小, 则必须在调入新字体控制命令中加入字体放大系数注解:

```
\newfont{新字体转换命令}{TFM 文件名 scaled\magstep$n$}
```

这里 $n$ 可以为“half”或1到5之间的整数, 它们分别表示将原字库中的字体放大 $\sqrt{1.2}$ 、1.2、 $1.2^2$ 、 $1.2^3$ 、 $1.2^4$ 和 $1.2^5$ , 其放大的效果相当于(如果调用了10pt的字库文件)分别使用了以下字号转换命令:

```
\large \Large \LARGE \huge \Huge
```

需要缩小时, 则必须在控制命令中加入字体缩小倍数注解:

```
\newfont{新字体转换命令}{TFM 文件名 scaled 数字参数}
```

这里数字参数可取 $10000 \times \eta$ : 当 $\eta$ 取1时, 就是原来的尺寸; 当 $\eta$ 取小于1的数字时, 字体则相应地缩小; 当然 $\eta$ 取大于1的数字时, 即为放大。例如数字参数取9000, 打印字体的字号为9pt。

④ 由于计算机的基本内存有限, 用户不要同时在一个源文件中调入了过多的新字库(同时定义了过多的新字体命令), 否则系统会溢出。解决的办法之一是将已经定义的新字体转换命令在使用完毕后立即予以注销:

```
\renewcommand{新字体转换命令}{}
```

但有时仍会溢出, 这时, 最好将源文件分成几个小文件排版。

至于数学公式中使用的特殊符号, 由于没有简单的英文字母与之一一对应, 所以得使用特殊符号调用命令:

```
\symbol{n}
```

其中 $n$ 表示特殊符号字库中的第 $n$ 个字符。例如:

```
\newfont{\amsy}{amsy10}
{\amslasy \symbol{56} \symbol{57} \symbol{58} \symbol{59}}
```

可以得到如下打印结果:

$\forall \exists \sim \emptyset$

当然, 具体每个字符所对应的序号得从系统操作指南获得; 若无法找到, 则可将所有特殊符号统一打印出来, 以留待日后参考。



## 10.10 断行控制命令

一般情况下,  $\text{\LaTeX}$  系统对排版过程中正文版面断行和断页问题会考虑得非常周全。因此, 用户在排版过程中不必去顾及如何断行和断页的具体细节。但是, 在最终输出硬拷贝时, 如果用户对某些版面不够满意(例如有可能个别单词超出了右边界, 或者某页底部最后行没有撑足版面, 等等), 那么可以通过插入一些断行和断页命令来帮助  $\text{\LaTeX}$  作局部调整。

所谓断行是指不管正文中内容在排版时是否处于版心右边界, 采用人工干预的方法令系统将当前行正文内容与后续行文字断开(即无条件换行), 并且根据当前版面的边界将行文内容左右(所有单词)均匀地拉开, 从而使左右两边均能撑足版面。在前面章节中使用的换行控制符“ $\backslash$ ” (或换行命令“ $\backslash$ newline”)只是结束当前行的排版, 但并不要求行文撑足版面。因此, 本书中提到的断行与换行是不同的版面控制方法。

### 10.10.1 断字注解命令

通常情况下, 系统会自动处理行文排至版面右边界处断行的问题。在这个排版过程中主要牵涉到英文单词的拆字(断字)规则, 一般系统会处理得相当好。但系统有时会不知道如何对某些复杂单词, 这时候, 用户除了采用在可以进行断字的适当位置插入控制字符“ $\text{\textasciitilde}$ ”(参见2.3.5节)之外, 还可在全局说明区域对使用频率较高的单词作些注解。这种断字注解命令的格式为

```
 $\backslash$ hyphenation{字符串1-字符串2-...-字符串n}
```

其中“-”是指可以断字的位置。例如对单词“formulas”, 用户可在“ $\backslash$ begin{document}”命令之前输入:

```
 $\backslash$ hyphenation{for-mu-las}
```

这样, 注解命令给系统提示了可断字的两个假定位置,  $\text{\LaTeX}$  在遇到单词“formulas”时就会知道如何选取合适的位置拆字断行了。

### 10.10.2 断行控制命令

有时, 排版接近行尾时恰好遇到一个较长的但不能拆开的LR排版盒子(版面上的排版效果很可能是盒子中的文字挤出了右边界线), 这时用户可以在盒子前面或其它适当位置插入以下断行控制命令:

```
 $\backslash$ linebreak
```

于是, 再次排版时, 系统会将命令后面的文字换行起排, 同时将当前行文字根据版心边界两边拉齐。下面是一个使用“ $\backslash$ newline”和“ $\backslash$ linebreak”两条控制命令的排版实例:

```
A \verb|\newline| command ends a line
without justifying it. \newline But the \verb|\linebreak|
```

command causes the system to justify the line, `\linebreak` stretching the space between words so the line extends to the right margin.

排版结果:

A `\newline` command ends a line without justifying it.

But the `\linebreak` command causes the system to justify the line, stretching the space between words so the line extends to the right margin.

如果用户在命令“`\linebreak`”后面添加一个置于方括号中的参数,那么可以将无条件断行命令转变成程度不同的请求系统断行的控制命令:

`\linebreak[n]`

其中 $n$ 必须是取0到4间的整数。数字越大,则意味着对断行的要求程度越强烈。当取值为4时,即等价于无参数的断行命令“`\linebreak`”。而取0时,则表示,插入命令的地方可以作为断行位置,其请求系统断行的程度最不强烈,既非促使,亦非阻拦系统进行断行。

如果用户对系统已经作断行处理的位置不满意,那么还可以使用阻止断行的控制命令:

`\nolinebreak[n]`

其中 $n$ 同样是表示劝阻断行的强烈程度。

### 10.10.3 自然段断字处理环境命令

如果用户对某一自然段(用空行或命令“`\par`”隔开)的排版结果(主要是指断字)不太满意,然而又不知如何处理断字和断行问题,那么可设置一个比较有效的令系统对整段文字通盘考虑的版面断字处理环境命令。

```
\begin{sloppypar}
自然段正文
\end{sloppypar}
```

此环境命令有一个功能相似的自然段断字控制命令“`\sloppy`”,使用时,可另外用分组控制符设置自然段的边界:

```
{\sloppy 自然段正文}
```

在此命令环境中,系统排版时,主要考虑的是如何将版面右侧拉齐,而不会过分去关心字词的间距。所以有可能出现单词间的空距较大的排版效果。

## 10.11 断页控制命令

所谓断页是指不管正文中内容在排版时是否处于版心底部, 采用人工干预的方法令系统将正文内容与后续文字断开(即无条件换页), 并且根据版式定义的版心边界将行文内容上下(所有行) 均匀地拉开。在用户不满意系统排版处理的版面时, 可使用以下形式的断页控制命令:

`\pagebreak[n]`

其中 $n$  的意义与断行命令“`\linebreak[n]`” 中的相类似, 反映了用户要求断页的强烈程度。注意另页(面) 命令“`\newpage`” 与“`\pagebreak`” 的不同之处在于前者换页时不会将当前版面撑满。

如果用户不希望系统在某个位置断页, 那么还可以使用阻止断页控制命令:

`\samepage`

`\nopagebreak[n]`



## 11

# 信件类 (letter) 版式和 APS 版式

本章首先介绍 $\text{\LaTeX}$ 系统定义的信件类(letter)版式。作为特殊类型的版式实例,本章着重介绍一个已经标准化可在 $\text{\LaTeX}$ 排版系统中运行的美国物理协会APS科技论文版式。

### 11.1 信件类版式

除了文章类(article)、报告类(report)和书籍类(book)文件版式之外,比较常用的就是打印公务或私人信件(letter)的文件版式了。这种文体类型可以使排印西文信件变得非常容易:用户不必去考虑如何安排寄信人或收信人的地址排版位置,也不必去为署名和日期位置的具体细节而费心;此外,用户可以在一个源文件中同时编排若干封不同的信件,并且可在输出时生成每个信封上粘贴用的地址标签。

#### 11.1.1 信件类(letter)文件版式命令

根据标准的 $\text{\LaTeX}$ 程序语言设计要求,除了编制信件类版式所需的一些特殊控制命令和排版环境之外,源文件中必须要有如下包含信件文件版式在内的排版控制命令:

```
\documentstyle[版式注解]{letter}
%全局说明区域控制命令
\begin{document}
%发信人有关通讯信息
%信件1)
%信件2)
%.....
\end{document}
```

版式注解一般可用于对正文基准字号加以说明,例如“12pt”。在信件版式的全局说明区域内,用户可自行对版心尺寸、行距等版面加以特殊安排。

### 11.1.2 发信人有关通讯信息

在用户(写信人)同时向各单位或个人发出的信件中,总需要重复写上完全相同的有关发信人通讯信息(例如写信人姓名、地址、电话等)的文字部分。所有这些内容可以通过在文稿排版起始命令`\begin{document}`后面选择下列带单参量的控制命令来一次实现:

- `\name{写信人姓名}`

此命令用于向系统输入用户(写信人)的姓名信息,  $\text{\LaTeX}$  将会在后面所有信件中打印上可供回信者参考使用的寄信人姓名。例如:

```
\name{Wei-guo FENG}
```

- `\address{写信人地址}`

此命令用于控制编排用户(写信人)的详细地址,  $\text{\LaTeX}$  将会根据西文信件的形式,在后面所有信件中的标准位置上打印出写信人地址。对于需要分成多行的详细地址,用户可用控制符“`\\`”将其隔开。例如:

```
\address{Department of Physics \\ Tongji University \\
          1239 Siping Road \\ Shanghai 200092 \\ P. R. China}
```

对于使用规范的印有单位地址等详细信息的信纸场合,此命令可省略。省略时系统将在天头位置留出适当位置,以保证正文不与信纸上方的单位地址重叠。

- `\signature{写信人姓名及头衔}`

此命令用于向系统输入写信人的姓名及头衔信息,  $\text{\LaTeX}$  将会在后面所有信件的末尾打印上寄信人姓名及头衔,并留出适当的供寄信者签名的空白位置。对于需要分成多行的详细衔头,同样可用控制符“`\\`”将其隔开。例如:

```
\signature{Wei-guo FENG \\ Ph. D and
            \\ Associate Professor}
```

- `\location{位置}`

对于机构比较庞大的单位,例如高等院校,工矿企业,前面的通讯地址还不足以确定寄信者的办公室位置,这时,还可以用此命令来确定详细方位。比如大楼名称,房间号码等。例如

```
\location{206 Lihua Building}
```

- `\telephone{电话号码}`

向排版系统输入用户电话号码。例如

```
\telephone{(021)-545-5080 Branch 3796}
```

以上命令(除了“\address”和“\signature”命令之外)有些可能在低版本的 $\text{\LaTeX}$ 排版系统中无效,这时,用户可将有关内容合并和其它项目之中。例如将电话号码用“\”隔开后合并在地址参量中。另外,这些控制命令在源程序中的先后顺序对输出时的规范排版样式没有影响。由于一般微机DOS系统自动设有日期和时钟(用户可作更改),所以信件中的日期是由 $\text{\LaTeX}$ 自动编排和打印的。

### 11.1.3 信件排版环境

每封信均由一个带收信人地址参量的信件排版环境命令来控制,其标准形式为:

```
\begin{letter}{收信人姓名和地址}
<信件正文>
\end{letter}
```

其中收信人地址同样可用控制符“\”来分成若干行排版。当各封信内容编排好以后,如果需要在信件排版输出时自动打印出所有收信人的姓名和地址标签,则只要在全局说明区域再插入一个生成标签控制命令

```
\makelabels
```

就可以了<sup>1</sup>。

### 11.1.4 信件正文的引出和结尾

差不多世界上所有信件的正文均是以对收信人称呼(例如“亲爱的……”)开头,并以礼仪性客套用语(例如“此致敬礼”)收尾的。 $\text{\LaTeX}$ 在处理这些版面是通过以下两个命令来实现信件正文的引出和结尾排版功能的:

```
\opening{信件起始称谓用语}
.....
\closing{信件结尾礼仪性用语}
```

例如:

```
\opening{Dear Professor John,}
.....
\closing{Best regards}
```

系统将自动把两个控制命令中的文字参量分别编排在信件的起始和结尾位置,并在正文下方自动打印出由命令“\signature”定义的寄信人姓名和头衔。夹在上述两命令之间的所有西文文字将按通常的行文状态排版。但要注意,许多在其它文体类型中使用的复杂的排版控制命令(如章节命令等)不再有效。

<sup>1</sup>在国外,市场上已有A4纸大小(分隔成3×11小张)的地址标签专用粘贴纸供应, $\text{\LaTeX}$ 系统控制输出的地址标签就是对这种特殊纸张设计的。

### 11.1.5 信件附加信息

在信件正文结束之后(在命令“\closing”后面), 根据信件内容需要, 用户还可以在版面上添加一些信息, 如副本邮寄对象、附件和附言等。以下列出了可供用户选择的有关附加信息的控制命令:

- \cc{另外的收信人姓名}

此命令用于输出通常由“cc:”形式引出的所有收到该信件副本的收信人姓名。相邻的收信人姓名之间可用控制符隔开。例如:

```
\cc{J. Tinkers \ R. Evers \ C. Chance}
```

- \encl{附件}

用来指明该信中还有其它的一些附件。例如:

```
\encl{Three copy of the manuscript}
```

- \ps

此命令没有参量, 主要是用来产生可让寄信人书写附言的版面。如果需要拓宽版面, 可使用行空白间距命令, 例如“\medskip”或“\bigskip”。

### 11.1.6 信件排版实例

综上所述, 一个完整的 $\text{\LaTeX}$ 信件源文件应按照如下形式编写:

```
\documentstyle[版式注解]{letter}
  <全局说明区域控制命令>
\makelabels
\begin{document}
\name{写信人姓名}
\address{写信人地址}
\location{位置}
\telephone{电话号码}
\begin{letter}{收信人姓名和地址}
\opening{信件起始称谓用语}
.....
\closing{信件结尾礼仪性用语}
\cc{另外的收信人姓名}
\encl{附件}
\ps
\end{letter}
  <信件2>
```



.....  
`\end{document}`

## 11.2 APS 科技论文版式

美国物理协会(American Physical Society) 为了加快由该协会主办的各种物理类杂志的出版速度, 推出了一套可在 $\text{\LaTeX}$  排版系统中运行的APS 科技论文版式宏程序库软件。软件所定义的文件输出硬拷贝将完全符合物理类权威性杂志 *Physical Review* (《物理学评论》) 及 *Physical Review Letter* (《物理学评论快报》) 的规范要求<sup>2</sup>。因此, 凡向美国物理类科技杂志(特别是《物理学评论》) 寄送的科技论文稿件审稿通过后, 可直接寄送(既可通过邮局以软盘形式邮寄, 也可通过电子通讯e-mail 方式传送) 按APS 版式编写的科技论文源程序, 这样可大大缩短论文排印发表的时间。

### 11.2.1 APS 宏程序库文件的结构和安装

APS 科技论文版式宏程序工具库主要包括以下软件:

README	APS 工具库文件结构和安装说明
APGUIDE.TEX	APS 科技论文版式编程指南“ <i>Input Guide for <math>\text{\TeX}</math>Author-Prepared Compuscripts</i> ”
APS.STY	aps 文件格式宏定义库文件
APS10.STY	以出版校样形式输出的宏定义库文架
REVTEX.STY	revtex 版式注解宏定义库文件
PREPRINT.STY	preprint(预印本样式输出) 版式注解宏定义库文件
SMPLEA.TEX	APS 版式简单源文件实例
SMPLEB.TEX	以《物理学评论》标准样式排版的完整论文源程序
SMPLEC.TEX	论文内容与文件“SMPLEB.TEX” 相同, 但在输出时以预印本样式排版。

以上软件由美国《物理学评论》出版社免费提供, 用户可直接去函索取最新版本的APS 版式软件。

安装APS 版式软件时, 必须将四个扩展名为“.STY” 的宏程序文件拷贝到用户自己的 $\text{\LaTeX}$  排版系统“STY” 宏程序库子目录中。这个子目录通常取名为“TEXINPUT”。至于其它的文件可复制到 $\text{\LaTeX}$  排版系统的当前目录中。由于所有的TEX 文件均是可执行排版源程序, 所以用户可在系统中逐个运行 $\text{\LaTeX}$  来获得硬拷贝。

<sup>2</sup>参见《物理学评论》撰稿指南 *Physical Review Style and Notation Guide*

### 11.2.2 APS 文件版式的环境设置

APS 版式实质上是根据 *Physical Review* 杂志的特点, 对 $\text{\LaTeX}$  排版系统中的文章类(article) 文件格式在部分排版功能上作了一些调整, 此外, 对个别 $\text{\LaTeX}$  排版命令在输入形式作了局部改动。根据输出硬拷贝的不同要求, APS 版式的源程序文件形式有两种: 预印本样式和 *Physical Review* 杂志校样形式。

#### 一、预印本样式

```
\tolerance = 10000
\documentstyle[preprint,revtex]{aps}
\begin{document}
\draft
.....
\end{document}
```

在输出时将以预印本样式打印。这是一种杂志社要求的送审稿件的打印样式。正文基准字号比出版时的校样略大一些(12pt), 版心宽度数值也比较大(15.5cm), 行距设置成为隔行打印形式, 以供编辑在文稿中作行间批注。最早版本的APS 系统(Version 1.0) 中没有将预印本样式设置成严格的如同打字机上隔行打印的形式。这时, 用户可以在全局说明区域内插入如下控制命令:

```
\def\baselinestretch{1.5}
```

#### 二、*Physical Review* 杂志校样形式

```
\tolerance = 10000
\documentstyle[revtex]{aps}
\begin{document}
\draft
.....
\end{document}
```

在输出时将以正式出版的样式打印。这是 *Physical Review* 杂志排版的校样形式。除较长的特殊数学公式和图表之外, 一般正文排印时的版面宽度比较窄, 这样可以将输出的拷贝拼嵌成双栏形式照相制版。对于正文中没有宽行数学公式、长形图表的情形, 用户可在版式注解参量中添加“twocolumn”, 即

```
\documentstyle[twocolumn,revtex]{aps}
```

这样, 就可获得排成双栏形式的版面了。

### 11.2.3 APS 版式的尺寸控制

对于 *Physical Review* 杂志排版的校样形式, 用户可在源程序中使用以下三条无参量的尺度控制命令来适时调节版心宽度:

`\narrowtext`      `\mediumtext`      `\widetext`

如果在源程序中插入前者, 将控制后续的正文以  $8.7\text{cm}(3\frac{3}{8}\text{in})$  宽度的窄条形版面排版, 而后两条命令则将后续的正文分别转换到以  $14\text{cm}(5.5\text{in})$  和  $18\text{cm}(7\text{in})$  宽度排版(通常在编排长数学公式或较宽的图表时使用)。

说明:

① 注意在某些特殊版面使用了后两条命令之后, 不要忘记再使用 “`\narrowtext`” 命令返回到窄条形版面排版状态。

② 上述版心宽度控制命令在以预印本样式排版输出时不产生任何作用。系统会自动将预印本样式的版面宽度设置为  $15.5\text{cm}$ 。

③ 不要在 APS 版式的源文件中使用 “`\smallskip`”、“`\bigskip`” 或其它任何形式的  $\text{\LaTeX}$  纵向或横向空距命令。

### 11.2.4 题名版面

根据 *Physical Review* 杂志的出版规范, 论文题目、作者署名及研究单位等有关题名部分版面的控制命令与  $\text{\LaTeX}$  题名排版命令相比, 有明显差别。在  $\text{\LaTeX}$  标准排版命令中, 题名部分的版面是通过全局说明区域的定义语句与正文中的题名排版命令合并使用来完成排印的(见第 5.7 节)。在 APS 版式中, 题名部分的版面无须在全局说明区域内作任何注解, 用户可在正文中直接使用由 APS 版式宏程序库定义的特殊控制命令。

#### 一、题目排版环境命令

APS 版式宏程序库为论文题目的排版设置了专用的环境控制命令:

```
\begin{title}
.....
\end{title}
```

在标题排版环境中, 系统会默认用户在输入长标题时用 “Enter” 键敲入的硬回车符作为换行标识符, 即源程序中的标题无须双反斜杠换行符 “`\\`” 来控制分行。系统将按照源程序中的自然分行形式逐行居中排版。

#### 二、作者署名排版命令

作者署名部分的版面由两条命令来控制。对第一作者(或在同一个研究单位的作者), 排版命令为:

`\author{作者署名}`

而对其余的合作者(第二、第三或在其它服务单位的合作者群体), 排版命令为:

`\moreauthors{作者署名}`

说明:

① 作者署名可以是一个或者几个, 视服务单位的差别而定。合作者署名之间可用逗号“,”或“and”分开。

② 命令“`\author`”只能使用一次, 用于编排第一作者或第一研究单位的合作者群体的署名版面。

③ 命令“`\moreauthors`”可使用若干次, 用于编排其余合作者或所有其他研究单位合作者群体的署名版面。

### 三、研究单位名称排版命令

对所有作者, 在署名版面后面都可使用以下环境命令来控制研究单位名称的排版:

`\begin{instit}`

.....

`\end{instit}`

在研究单位排版环境中, 系统亦会默认用户在输入较长的单位全称时用“Enter”键敲入的硬回车符作为换行标识符, 即源程序中无须双反斜杠换行符“`\\`”来控制分行。系统将按照源程序中的自然分行形式逐行居中排版。

### 四、收稿日期排版命令

杂志社接到论文稿件的日期控制命令为:

`\receipt{收稿日期}`

收稿日期这行文字通常是在论文正式出版时由编辑插入。

### 五、论文摘要排版环境命令

论文摘要部分的排版环境命令形式与标准的 $\text{\LaTeX}$ 命令相同, 格式为:

`\begin{abstract}`

.....

`\end{abstract}`

只是在输出时根据APS版式要求将黑体字标题“Abstract”省略了。

## 六、PACS 分类索引码排版命令

为了实现编制索引的规范化, 凡属物理学类或天文学类的科技论文, 在 *Physical Review* 杂志上正式发表时均需在摘要下面标记 PACS (Physics and Astronomy Classification Scheme) 分类索引码。编排 PACS 索引码的控制命令为

`\pacs{PACS 索引码}`

PACS 索引码的划分取决于论文的具体内容, 用户可查阅每年出版的最后一期 *Physical Review Letter* (《物理学评论快报》), 以找到与自己论文合适的 PACS 索引码。

下面是 APS 版式工具库中文件“SMPLEA.TEX”中有关题名部分排版的源程序:

```
\tolerance = 10000
\documentstyle[revtex]{aps}
\begin{document}
\draft
\begin{title}
Title of manuscript:
One line of input produces one line of output
\end{title}
\author{A. A. Author and B. B. Author}
\begin{instit}
Authors' institution and/or address
One line of input produces one line of output
\end{instit}
\moreauthors{C. C. Author}
\begin{instit}
Second author institution and/or address
One line of input produces one line of output
\end{instit}
\receipt{January 1 1989}
\begin{abstract}
The author will not know the received date when the
compuscript is first submitted; production will insert
this. Every article includes an abstract.
The abstract is a concise summary of the work covered
at length in the main body of the article.
It is used for secondary publications and for information
retrieval purposes. Valid PACS numbers should be entered
after the abstract is finished, using the
\verb|\pacs| command.
```

```

\end{abstract}
\pacs{Valid pacs appear here.}
\narrowtext
.....
\end{document}

```

不管是以预印本样式打印还是以 *Physical Review* 杂志校样形式打印, 题名部分的版面均将以通栏形式(版心宽度为差不多覆盖整个A4 纸版面)。

### 11.2.5 节次标题

在 *revtex* 版式注解宏定义库文件—*REVTEX.STY* 中定义了三个层次的节次排版命令, 其形式与标准的 *L<sup>A</sup>T<sub>E</sub>X* 节次排版命令完全一样(参见本书第5.1 节)。所有节次标题均采用居中样式排版(标准 *L<sup>A</sup>T<sub>E</sub>X* 宏程序库定义的标题版式是靠左边界排版的)。第一层次的节标题(由命令“*\section*”控制), 在排版时, 所有字母将被转换成大写形式(采用西文黑体字和正文基准字体的字号), 而节次序号将以罗马数字字体形式打印; 第二层次的节标题(亦由命令“*\subsection*”控制) 采用小一号黑体字并按原标题字母形式输出, 序号采用英文大写字母; 第三层次的节标题(由命令“*\subsubsection*”控制) 用小一号斜体字印出, 序号采用阿拉伯数字。至于更低层次的标题得由用户根据需要按文献杂志的规范自行定义了。

### 11.2.6 脚注和参考文献

根据 *Physical Review* 杂志规范, 脚注和参考文献合并编排。类似于标准的 *L<sup>A</sup>T<sub>E</sub>X* 参考文献排版环境命令, APS 版式的脚注和参考文献采用如下排版环境命令:

```

\begin{references}
\bibitem[脚注序号1]{脚注1 关键词} (脚注1 正文)
\bibitem[脚注序号2]{脚注2 关键词} (脚注2 正文)
.....
\bibitem[文献序号1]{文献1 关键词} (文献1 正文)
\bibitem[文献序号2]{文献2 关键词} (文献2 正文)
.....
\end{references}

```

说明:

① 在每条脚注或参考文献的正文最前面的方括号中的序码<sub>*i*</sub> 是指用户在自己选取的计数方式中的序码字符串。对脚注(如作者常住地址等) 用户应在方括号内依次选择以下标准的脚注序码专用控制命令:

```

* \dag \ddag \S \** \dag\dag \ddag\ddag \S\S

```

而对参考文献, 则应使用阿拉伯数字作为序码。这部分内容可省略, 省略时系统将用阿拉伯数字为脚注和文献统一编号。

② 标识每条脚注或文献的关键词 $i$ 的功能与 $\text{\LaTeX}$ 的引用标识符命令中设置的关键词功能相同。这里只是省去了控制令“ $\backslash label$ ”而已。

③ 排版输出时, 不会出现黑体字标题“References”, 而只是在与正文之间用一条直线分隔开, 并且所有的序码以上标形式排印(无方括号)。

④ 在正文中需要调用脚注或文献时, 可使用标准的 $\text{\LaTeX}$ 命令:

$\backslash cite\{ \text{关键词}i, \text{关键词}j, \text{关键词}k, \dots \}$

在排版输出时, 脚注或文献的序码同样以上标形式排印(无方括号)。

### 11.2.7 插图与表格版面的设计

按照论文送审的样式规范, 图表通常被编排在源程序的最后面(与正文分开), 以后由杂志出版社将图表嵌入正文之中。

#### 一、插图与表格标题的排版命令

为了与表格标题的版面区分开, 插图部分的标题及注文不再采用标准的 $\text{\LaTeX}$ 命令“ $\backslash caption$ ”, 而是使用APS版式的专用插图标题命令:

$\backslash figure\{\text{插图标题}\}$

在排版输出时, 插图标题前自动加上“FIG.”与插图序号(阿拉伯数字)。至于表格部分的标题排版命令仍然使用“ $\backslash caption$ ”。输出时, 表格标题前会被加上“TABLE.”和以罗马大写数字字体形式打印的序号。

需要交叉引用时, 对插图, 可以在标题注文参量中嵌入带参量标识符命令“ $\backslash label$ ”。而对表格, 则可在制表排版环境(tabular)退出时即插入交叉引用标识符命令。

#### 二、制表命令

APS版式中可划线的制表环境命令与通常的 $\text{\LaTeX}$ 可划线表格排版环境命令形式上完全一样,

```
 $\backslash begin\{tabular\}\{ \langle \text{列参数}1 \rangle \langle \text{列参数}2 \rangle \dots \langle \text{列参数}n \rangle \}$ 
 $T_{11} \quad \& \quad T_{12} \quad \& \quad \dots \quad \& \quad T_{1n} \quad \backslash \backslash$ 
 $T_{21} \quad \& \quad T_{22} \quad \& \quad \dots \quad \& \quad T_{2n} \quad \backslash \backslash$ 
.....
 $T_{m1} \quad \& \quad T_{m2} \quad \& \quad \dots \quad \& \quad T_{mn}$ 
 $\backslash end\{tabular\}$ 
```

其中 $T_{ij}$ 为表格中的第 $i$ 行第 $j$ 列元素。此命令可嵌套在活动表格排版环境table中使用。

然而从排版功能上相比, 两者有明显差别。首先, 在 $\text{\LaTeX}$  表格命令中, 划横线(水平线) 是通过命令“ $\backslash hline$ ”来控制的, 其长短取决于表格中的信息量多少。在APS 版式中, 命令换成

$\backslash tableline$

并且, 直线总是在版面上左右撑足(其长短完全取决于版心的宽度)。其次, 在进入APS 版式的表格环境或退出环境之前, 用户不必添加表格顶部和底部的横直线。系统会自动在表格顶部和底部分别加印双横直线。注意, 根据 *Physical Review* 杂志规范, 表格中应避免用竖线。再则, 在APS 版式可划线表格制表环境结束后, 某些项元素需要附加说明时, 可分别使用表格脚注命令:

$\backslash tablenotes$ { 脚注正文}

排版时, 表格被系统视作小页版面, 脚注将排在表格的底部, 序码由用户自行选取英文小写字母以上标形式编排(引用脚注的地方由用户自行添加上标形式的序码<sup>3</sup>)。

### 11.2.8 数学公式的编排

数学公式可按照标准的 $\text{\LaTeX}$  命令来进行编排。如果是编排打印正式出版的校样, 那么在处理较长的数学公式时可使用“ $\backslash mediumtext$ ”和“ $\backslash widetext$ ”来调节方程式所处的版面宽度(数学公式)。

除了使用标准的 $\text{\LaTeX}$  数学公式排版命令之外, 用户还可使用APS 版式中定义的以下几个数学公式排版命令。

#### 一、数学公式的编号

根据实际需要, 用户可以使用方程式自定义编号命令:

$\backslash eqnum$ { 自定义编号}

例如源程序

```
 $\begin{equation}$ 
 $f(x)=x^{\{2\}}$   $\backslash eqnum\{55\}$ 
 $\end{equation}$ 
```

可将方程 $f(x)=x^2$  的编号设置为“(55)”。

如果用户希望方程式的编号根据节次来排序, 那么可在文稿前面加注以下命令:

$\backslash math-with-secnams$

这样, 用户就可得到类似(1.1), (1.2), ...; (2.1), (2.2), ... 形式的方程序码号。

<sup>3</sup>命令“ $\backslash tablenotes$ ”不能自动编号。



## 二、数学公式的对位控制

对于可用于出版的校样, APS 版式定义的数学公式通常是缩进(约1cm)排版。如果在校样输出时需要将某个独立方程式顶格排版, 那么用户可在该方程式的数学状态起始命令之前(如`equation`, `eqnarray`, `displaymath`)使用数学公式顶格排版命令“`\FL`”。例如

```
\FL
\begin{equation}
{\cal F}_{i}=p_{1}-q_{i}
\end{equation}
```

校样输出时, 带有顶格排版命令“`\FL`”的数学公式将靠版心左边界排版, 而所有未添置顶格排版命令“`\FL`”的数学公式将缩进排版。注意此命令对预印本的排版过程无效(所有独立方程式仍将居中排版)。

## 三、分式命令

除了本书介绍的分式命令“`\frac`”和“`\over`”之外, APS 宏程序库亦定义了一个分式命令:

```
\case{分子表达式}/{分母表达式}
```

其排版功能与 $\text{\LaTeX}$ 分式命令“`\frac`”相同, 只是在命令输入形式上增加了一个表示分数线的ASCII字符“/”。



## 12

# 源程序语法错误的发现和处理

本章介绍如何发现定位错误, 以及几种常见的错误信息或警告信息的含义和处理方法, 并且详细列举了常见的出错信息。

### 12.1 出错定位

$\text{\LaTeX}$  系统在执行排版时, 如果没有任何语法错误, 则会顺利结束; 如果  $\text{\LaTeX}$  发现源文件中有严重的语法错误, 就会立即停止运行, 并在屏幕上显示发现错误的位置以及有关的出错信息; 如果在排版过程中, 源程序中缺少某些控制参数、版面处理欠妥或者某些事先安排的排版过程无法实现, 则系统亦会显示警告信息, 但不会停止排版。在大多数情况下, 耐心的用户可以根据屏幕显示的有关出错定位信息, 仔细对照自己编制的源程序, 就能发现出错的原因。当然, 查错的速度快慢与用户使用  $\text{\LaTeX}$  系统的经验多少直接相关。从某种程度上讲, 如果用户能熟练地处理排版过程中的各种错误, 那么  $\text{\LaTeX}$  便会给予高质量的输出硬拷贝作为奖赏。

$\text{\LaTeX}$  系统为用户显示错误位置的方法是将它认为源程序中有语法错误的那一行的行号以及与错误内容有关的部分正文显示出来。如“1.78”则表示源程序中的第78行可能有语法错误(前面的字母“1”是英文单词“line”(行)的缩写)。不过屏幕上有时显示的错误信息会令用户不知所措。例如排版过程中屏幕上时而会出现一些冗长的难以阅读的控制命令的字符串。在这些信息中, 有些完全是多余的。出现这些信息主要是因为  $\text{\LaTeX}$  系统是借助于另外一个较复杂的  $\text{\TeX}$  排版系统来定义的宏命令库。因此, 许多信息实质上就是  $\text{\TeX}$  排版系统缺的语法错误或警告信息。在  $\text{\LaTeX}$  系统为用户执行排版的同时,  $\text{\TeX}$  系统也在用它自身的查错体系检查用户的源程序。当它认为是排版错误时, 就命显示  $\text{\TeX}$  系统错误信息。在许多场合下, 只要系统没有停止运行任务, 用户可以暂时不必去管这些信息。

如果用户实在无法根据屏幕显示的信息找出产生严重错误的原因, 那么还可以由屏幕显示的信息来查阅本书附录D中有关出错信息的详细解释。

## 12.2 T<sub>E</sub>X 出错信息

首先让我们来看一个排版程序实例(其中有意加进了一个未经定义过的非法排版命令“\wrong”):

```
\documentstyle{article}
\begin{document}
This is an example of the \wrong (invalid) control command.
\end{document}
```

当用户对此文件执行排版时, 在屏幕为你显示若干常规的排版信息之后, 便会显示如下出错信息:

```
! Undefined control sequence.
1.3 ... This is an example of the \wrong
                                           (invalid) control command.
?
```

此刻, 系统暂时停止排版, 并将计算机光标停在问号“?”后面。在这个例子中, 出错信息是比较直接了当的。第一行以感叹号“!”引出的文字是T<sub>E</sub>X 错误信息的标准显示格式(在附录D中列举了常见的一些T<sub>E</sub>X 错误信息)。显示的主要内容是指出排版程序中语法错误的类型。第二行行首的记号“1.3”是指错误出现在源程序的第三行, 当然它同时也告诉用户系统在读到此错误之前已经完成排版的源程序总行数。由于命令“\wrong”不是一个L<sup>A</sup>T<sub>E</sub>X 系统的合法排版命令, 所以导致了一条错误信息。

如果此时用户需要帮助, 可以再键入一个问号“?”, 系统立即给出如下所示的选择菜单:

```
Type <return> to proceed, S to scroll future error messages,
R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore the next 1 to 9 tokens of input,
H for help, X to quit.
```

其中用户可选择输入的各项指令的功能如下:

- <return> (回车键): 按回车键后, L<sup>A</sup>T<sub>E</sub>X 系统将尽最大努力从错误中恢复并继续执行排版。如果新的语法错误出现, 系统又会暂停。在不清楚产生错误的原因时, 用户可以连续使用此指令来完成排版, 然后再从输出拷贝中寻找出错来源。
- S (继续运行): 输入字母“S”(大小写无关, 以下同)即可令系统继续运行。即使后续的源程序中还有其它语法错误, 系统也不会停止排版。在排版过程中, 屏幕上将连续显示系统读到的每条语法错误的有关信息。对于运算速度较快的计算机,

屏幕显示的有关错误信息会一闪而过, 用户可能看不清除。不过这并不要紧, 因为系统会将所有出错信息存放在一个引用名与源文件相同, 而扩展名为LOG 的文件之中。用户可以在排版完毕后通过阅读LOG 文件来查错。

- R (强行运行): 功能与“S”类似, 不过比指令“S”更强硬, 甚至连打不开文件这样一类严重错误也不会使排版过程终止。
- Q (“静态”运行): 功能与“R”类似, 不过屏幕上不再显示出错或提示信息。出错信息将直接写入“LOG”文件, 因此, 显示器相对比较“安静”。
- I (修改命令): 当显示屏幕指出源程序中出现了非法字符串(排版控制命令)以后, 用户可在问号“?”后面输入字符“I”, 并紧接着再输入一个正确的字符串(控制命令)。这样, 在用户按下回车键后, 系统便会将输入的字符串替换排版过程中发现的错误字符串。例如本节开始时给出的排版实例, 当系统读到非法命令“\wrong”时, 在屏幕上显示有关信息以后便停止运行。如果用户想用另外一个命令(例如“\LaTeX”命令)替代它, 则可在问号后面输入(用下划线标记)

? I\LaTeX

最后按下回车键, 系统便会以交互方式改正这个错误, 并继续执行排版。用户必须注意的是, 用指令“I”只能在排版过程中动态地修改错误, 虽然可以产生正确的输出硬拷贝, 但没有真正修改用户编制的排版源文件。要想保存正确的源文件版本, 还必须使用文本编辑软件对源文件作修改。

- n ( $0 < n < 10$ ) (删除命令): 在问号“?”后面输入1到9之间的任意一个整数n表示动态地删除从当前位置起n个字符或命令。系统会将删除后的样式反馈到屏幕上, 以让用户确认最后的结果是否正确。
- H (请求帮助): 输入字符“H”, 并按下回车键, 便可获得系统给出的帮助信息。例如, 在本节中列举的排版错误, 通过此指令, 便可获得如下帮助信息:

```
The control sequence at the end of the top line
of your error message was never \def'ed. If you have
misspelled it (e.g., '\hobx'), type 'I' and the correct
spelling (e.g., 'I\hbox'). Otherwise just continue,
and I'll forget about whatever was undefined.
```

用户可根据系统给出的建议, 或者用修改指令“I”作动态修改, 或者对源文件作永久性修改。对于常见的语法错误, 系统给出的信息很有用; 对于一些古怪的语法错误, 系统显示的内容对用户帮助不大。这时得靠用户自己来分析和处理语法错误了。

- X (退出系统): 此命令可令系统无条件停止运行。前面已经排版好的内容将输出至DVI文件中。

关于常见的TeX 系统出错信息详细解释可查阅附录D.2。

### 12.3 L<sup>A</sup>T<sub>E</sub>X 出错信息

现在让我们来分析一些稍微复杂一点的语法错误。在下面的引文(quote) 环境排版实例中, 进入环境的注解词的拼写发生了错误。

```
\documentstyle{article}
\begin{document}
\begin{quoth}
Here is the example of \LaTeX\ Error originated from
the undefined environment.
\end{quote}
\end{document}
```

在执行排版过程中, L<sup>A</sup>T<sub>E</sub>X 会在屏幕上显示如下错误信息:

```
LaTeX error. See LaTeX manual for explanation.
          Type H <return> for immediate help.
! Environment quoth undefined.
\@latexerr ...diate help.}\errmessage {#1}
                                          \endgroup

1.3          \begin{quoth}

?
```

这里, 语法错误是属于L<sup>A</sup>T<sub>E</sub>X 系统, 而不是TeX。字符串“LaTeX error.”引出的文字是L<sup>A</sup>T<sub>E</sub>X 出错信息的标准显示形式。类似TeX 出错信息, 后续的用“!”引出的文字是L<sup>A</sup>T<sub>E</sub>X 语法错误的简单解释。用户可以通过查阅在本书附录D.1 中的出错信息的详解来处理源程序中的语法错误。至于接下去的一行复杂的文字“\@latexerr ...”并不重要, 用户可不必去管这些字符串的具体意义。在问号后面同样可以选择键入一些指令。例如, 键入“H”和回车键以<sup>①</sup>获得帮助信息:

```
Your command was ignored.
Type I <command> <return> to replace it with another command,
or <return> to continue without it.
```

此时, 用户或者按回车键继续运行, 或者使用“I”(取代) 命令作动态修改。对上面例子, 用户可以通过键入指令

```
? I\begin{quote}
```

和回车键来对运行过程中的排版程序作暂时修改。对于常见的语法错误，用户可以根据出错信息对源程序作修改。如果一时无法进行修改工作，可以通过连续按回车键来完成排版任务。然后由输出拷贝来分析出错原因。例如，在拷贝中出现了连续几个页面的意大利斜体，这很可能忘记了分组符号“}”或是忘记恢复到罗马字体。如果出现的是很长篇幅的数学状态斜体字，则显然是忘记退出数学公式状态。例如，忘记了添加控制符“\$”或“\”。

## 12.4 语法严重错误

本节介绍几条常见的源程序语法严重错误，例如源文件中缺少文件版式命令或缺少文稿排版环境命令等。

### 12.4.1 无文稿排版环境

如果用户需要执行排版的源文件中既没有输入文件版式命令，也没有设置正文文稿排版环境命令，例如仅有一行英文文字

```
This file has no document header or any command else in it.
```

那么用户将获得如下 $\text{\LaTeX}$  出错信息：

```
LaTeX error. See LaTeX manual for explanation.
      Type H <return> for immediate help.
! Missing \begin{document}.
\@latexerr ...diate help.}\errmessage {#1}
                                         \endgroup
<to be read again>
      T
1.1      T
      his file has no document header or any command else in...
?
```

根据标准的 $\text{\LaTeX}$  规则，排版源文件的正文开始位置必须设置文稿起始命令“ $\text{\begin{document}}$ ”或者是输入文件命令“ $\text{\input}$ ”。在 $\text{\LaTeX}$  对上述例举的源文件执行排版时，由于最先读到的不是有效的文稿排版环境控制命令，所以立即停止了运行，并且将光标停在第一个字母“T”处。这时你可以令系统退出(按“X”键退出)。如果用户想要继续运行下去，那么系统会试图从终端读入另外一个文件；假如失败，只能通过按“Ctrl”键加上“C”或“Z”键来紧急中断运行。

### 12.4.2 无文件版式命令

如果用户在源文件正文之前的全局说明区域没有设置无文件版式命令，那么即使文稿起始和结束位置设置了排版环境命令，在 $\text{\LaTeX}$  运行时，仍无法执行排版任务。这

是因为在系统执行排版时还必须调入在文件版式命令中标识的有关文件(其中含有许多重要的 $\text{\LaTeX}$ 宏定义命令)。例如我们将上一小节中的源程序例子改成

```
\begin{document}
This file has no document style or any command else in
its preamble.
\end{document}
```

那么 $\text{\LaTeX}$ 在执行排版时, 会在屏幕上显示如下有关错误信息:

```
! Undefined control sequence.
\@floatplacement troom \bottomfraction
\@colht \global \@colnum \c
\document . \begingroup \@floatplacement
\@dblfloatplacement \makeat
1.1 \begin{document}
?
```

如果用户想要继续运行(按回车键), 那么系统会在屏幕上重复显示上述难以读懂的 $\text{\TeX}$ 出错信息, 这时用户应按“X”退出为宜。

### 12.4.3 无效文件版式

假定用户输入如下文件版式命令其中文体类型:

```
\documentstyle{wrongdocstyle}
```

其中文体类型参量为“wrongdocstyle”, 如果排版系统中无对应的扩展名为“[.sty]”文体类型宏定义文件, 那么你会在屏幕上发现 $\text{\LaTeX}$ 无法获得“wrongdocstyle.sty”文件的有关信息。这时系统会停下来, 并请求你输入另外一个文体类型宏定义文件名, 例如屏幕会显示:

```
! I can't find file 'wrongdocstyle sty'
<to be read again>
\relax
\@documentstyle {} \input #2 sty \relax
\let \@elt \input \@optionf.
1.1 ... \documentstyle{wrongdocstyle}

Please type another input file name
```

如果用户在光标后面输入正确的文体类型宏定义文件名(例如“article sty”), 那么系统就可以顺利运行下去, 否则, 系统又会在屏幕上重复显示与上述相类似的有关错误信息。如果要想退出系统, 可以采取强行中断措施(按“Ctrl”、“Z”键)。



#### 12.4.4 数学状态中的意外语法错误

有些ASCII字符(例如“\$”、“^”和“\_”等)在 $\text{\LaTeX}$ 系统中是作为数学公式排版控制符使用的。如果在行文本排版状态中误用了这些控制符,则会导致 $\text{\LaTeX}$ 无所适从,从而用户无法实现原先设想的排版效果。例如,在下面的排版源文件中,使用了一个美元“\$”符号:

```
This line is over $5.00 in price and also refers
to the variable foo_par which will cause a problem.
```

当 $\text{\LaTeX}$ 读到第一个美元符号时,立即会切换到数学公式排版状态。如果到一段行文本或整个文件的结尾一直没有出现退出数学公式的控制符,那么系统会认为源文件中少了一个与前面进入数学状态相匹配的美元符号。这时,系统暂时停止运行,并且在屏幕上显示以下错误信息:

```
! Missing $ inserted.
<inserted text>
          $
<to be read again>
          \par
.....
?
```

其中长省略号是指显示的一些复杂的 $\text{\TeX}$ 信息。假如用户按回车键继续,那么得到如下极其糟糕的排版效果:

```
This line is over 5.00in price and also refers to the variable foo.....
```

因为系统将后续的行文本全部视作一个方程式(所有空格均被忽略)。现在假定我们对源程序作一修改,在美元符号之前添上一个反斜杠控制符(打印美元符号),这样,源程序在下划线出现之前运行正常,但是 $\text{\LaTeX}$ 读到下划线时,又会停止运行。因为,下划线被系统保留在数学公式中作为下标排版控制符使用。如果出现下划线,则会被认为忘记使用控制符将排版状态切换到数学公式之中,因而,屏幕上会显示:

```
! Missing $ inserted.
<inserted text>
          $
<to be read again>
          -
1.4          to the variable foo_
                                par which will cause a problem.
?
```

如果用户按“h”寻求帮助,则可获得提示:

? h

I've inserted a begin-math/end-math symbol since I think  
you left one out. Proceed, with fingers crossed.

对上述情形, 系统认为用户忘记输入“\$”符号, 因此, 自动为用户设置数学状态; 如果用户需要继续下去, 那么后续的行文将又被视为数学公式的内容。前面提到的非预期的排版效果会重新出现。如果用户确实想要打印字符串“foo\_par”, 那么可使用“\verb”命令。

## 12.5 TeX 系统警告信息

在执行排版过程中, 虽然源程序中不存在任何语法错误, 但有时仍然会发生一些具体困难。例如某一行结尾处也许会出现一个较长的复杂单词, LaTeX 可能无法找到合适的断行点; 或者在某页中无法找到最佳的断页点。在这种情形出现时, 系统并不停止运行, 而只是在屏幕上显示一些警告信息(这些信息同时被复制在LOG 文件中)。由于对整个版面的排版过程并不造成严重错误, 在许多情况下, 用户可不必去管它。如果对版面的输出效果发现有不满意的地方, 可再作局部调整或修改。TeX 警告信息的区分比较容易, 它既没有“!”标识引出, 也没有用问号“?”来作为结尾, 只是简单地显示一行警告信息。

### 12.5.1 断行警告信息

在执行排版时, 最常见的处理断行的TeX 警告信息是形如

Overfull \hbox (xxx.xx pt too wide) ...

横向版面内容溢出信息, 或者是形如

Underfull \hbox ...

的警告信息。这两个信息的出现一般是由于系统在排某一行时无法找到恰当的断行点, 更准确的说, 是系统不知如何对位于行尾的长字符串单词作拆字处理。对前一种情形, 这一行版面输出时的总长度可能会超过原来设置的版心宽度; 而后一种情形, 则可能横向版面未撑满。

以上屏幕显示的内容只是一些警告信息, 系统并不会因此而停止运行, 所以用户不可能获得机会对版面作任何动态修改。如果用户发现某行内容输出时明显地超出了版面, 或者是版面未撑足, 则还可以通过对源文件作局部修改来进行调整。具体方法是:

- ① 对出现上述情形的原文作修改, 例如删除或增加一些单词。
- ② 在你认为可作拆字处理的合适位置设置一些控制符“\-”, 例如对于下列程序,

```
.....For a very interes\~ting account of the struggles of mankind
in the deter\~mina\~tion of the constants of physics, we recommend
Cohen {\em et al.}\~/: The Funda\~men\~tal Contants of Physics.....
```

L<sup>A</sup>T<sub>E</sub>X 除了可在它自己认为合适的位置作断行处理之外, 还可以选择在任何一个由用户设置的拆字控制符位置上打印连字符号, 并作换行处理。

③ 如果用户觉得需要告诉 L<sup>A</sup>T<sub>E</sub>X 如何对某个使用频率较高的单词作拆字处理, 还可以在全局说明区域使用拆字提示命令:

```
\hyphenation{...-...-...}
```

例如对单词“generation”, 你可在“\begin{document}”命令之前输入:

```
\hyphenation{ge-ne-ra-tion}
```

这样 L<sup>A</sup>T<sub>E</sub>X 在遇到单词“generation”时就会知道怎么办了。

④ 在超出版面的单词前面(或在未撑足版面的行尾单词后面)插入换行命令“\linebreak”。

⑤ 在出现警告信息的所在自然段(包括段与段之间的空行)设置 sloppypar 环境命令:

```
\begin{sloppypar} ... \end{sloppypar}
```

### 12.5.2 断页警告信息

在有些情形下, T<sub>E</sub>X 警告信息

```
Underfull \vbox (badness ...
```

是告诉用户系统没有发现最合适的断页点, 此页版面的高度很可能比定义的版心高度数值要小; 而在另一些场合下, 警告信息

```
Overfull \vbox (badness ...
```

同样是告诉用户系统没有找到合适的断页点, 此页版面的高度很可能比定义的版心高度数值要大。在第一种情形下, 由于系统为了撑满版面, 可能会在某些行之间引进额外的空白间距; 而在后一种场合, 则版面可能要比其它页面明显要高。如果用户觉得不合适, 可以使用另面排版控制命令“\newpage”或“\pagebreak”来人工干预版面。

## 12.6 L<sup>A</sup>T<sub>E</sub>X 系统警告信息

L<sup>A</sup>T<sub>E</sub>X 排版系统也有自身的警告信息, 通常屏幕显示的内容是以

```
LaTeX Warning: ...
```

形式引出。下面列出的是所有 L<sup>A</sup>T<sub>E</sub>X 系统的警告信息:

- Citation '...' on page ... undefined.

参考文献引用命令“\cite”中的标识符没有在“\bibitem”命令中定义过。当第一次对源文件执行排版时, 只要使用了文献的交叉引用, 就会出现上述警告信息。

这是因为AUX文件内还未记录下供文献交叉引用的有关信息。这时用户只要再运行一遍 $\text{\LaTeX}$ ，警告信息即会消失，如果系统仍然给出一些警告信息，则可对照检查引用标识符是否拼写出错。

- Label '...' multiply defined.

标识符“...”被“\label”或“\bibitem”多次使用。

- Label(s) may have changed. Rerun to get cross reference right.

由命令“\ref”、“\pageref”或“\”命令打印出来的数字可能会出错，因为最后一次运行 $\text{\LaTeX}$ 系统时，正确数字有可能被修改过了。

- Marginpar on page ... moved.

为了保证上下旁注之间保持一定的间距，某个旁注可能会在页面上向下移动了一段距离，这样就会导致与引用旁注的正文(插入“\marginpar”命令处的行文)不能对齐。

- Oval too small.

命令“\oval”中定义了一个比系统所能绘出的最小尺寸还要小的圆角矩形。输出时，用户可能得不到所期待的图形，排版结果可能不太美观。

- Reference '...' on page ... undefined.

交叉引用命令“\ref”或“\pageref”中的标识符没有在“\label”命令中事先定义过。当第一次对源文件执行排版时，只要使用交叉引用命令，就会出现上述警告信息。这是因为AUX文件内还未记录下供文献交叉引用的有关信息。这时用户只要再运行一遍 $\text{\LaTeX}$ ，警告信息即会消失，如果系统仍然给出一些警告信息，则可对照检查引用标识符是否拼写出错，或者事先根本就没有定义过。

## 附录 A

# 英文字体

- amr10

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 — Roman text

- amb10, ambx10

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 — Boldface extended

- anti10

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 — Text italic

- amu10

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 — Unslanted italic

- ans110

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 — Slanted roman

- ambxsl10

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 — Extended bold slanted

- amtex10, amtt10

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 --- Typewriter type

- amitt10

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 --- Italic typewriter

- amsltt10

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 --- Slanted typewriter

- amdunh10

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 — Dunhill style

• **ancsc10**

ABCDEFGHIJKLM  
 NOPQRSTUVWXYZ  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 0123456789 — SMALL CAPS

• **ambi10**

**ABCDEFGHIJKLM**  
**NOPQRSTUVWXYZ**  
*abcdefghijklmnpqrstuvwxy*  
 0123456789 — *Italic boldface*

• **anss10**

ABCDEFGHIJKLM  
 NOPQRSTUVWXYZ  
 abcdefghijklmnpqrstuvwxy  
 0123456789 — Sans serif

• **anssbbx10**

**ABCDEFGHIJKLM**  
**NOPQRSTUVWXYZ**  
*abcdefghijklmnpqrstuvwxy*  
 0123456789 —  
 Sans serif bold extended

• **anssi10**

*ABCDEFGHIJKLM*  
*NOPQRSTUVWXYZ*  
*abcdefghijklmnpqrstuvwxy*  
 0123456789 — *Sans serif italic*

• **anssq8**

ABCDEFGHIJKLM  
 NOPQRSTUVWXYZ  
 abcdefghijklmnpqrstuvwxy  
 0123456789 — Sans serif Q.

• **anssqi8**

*ABCDEFGHIJKLM*  
*NOPQRSTUVWXYZ*  
*abcdefghijklmnpqrstuvwxy*  
 0123456789 — *Sans serif Q. Italic*

• **anssmc10**

**ABCDEFGHIJKLM**  
**NOPQRSTUVWXYZ**  
 abcdefghijklmnpqrstuvwxy  
 0123456789 — Sans serif M. C.

• **amsy10** (\symbol{65}... \symbol{90})

*ABCDEFGHIJKLM*  
*NOPQRSTUVWXYZ*  
 — *Math calligraphic letter*

## 附录 B

# CCT 中西文排版系统简介

附录B将简单介绍CCT中西文排版系统的使用方法。内容主要包括CCT系统的总体结构、 $\text{C}\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 排版命令、设备驱动程序和图像接口等。

鉴于 $\text{T}_{\text{E}}\text{X}$ 软件以及由其衍生的排版软件(如 $\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ )在国际上已经成为标准的排版软件,在过去的十年中,世界上许多致力于微机排版系统开发的计算机科学家们一直在努力用自己的民族语言为 $\text{T}_{\text{E}}\text{X}$ 软件增添新的版本。例如国际上最新推出的集英语和德语两种语言为一体的 $\text{E}\text{M}\text{T}_{\text{E}}\text{X}$ 排版系统,就是这方面的成功典范。用户只要在规范的 $\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 源文件中调入有关处理德语字母的系统宏定义文件,就能非常容易地编排德语的书刊文件。

近年来,为了使我国的出版事业早日实现现代化,计算机专家们也一直在为 $\text{T}_{\text{E}}\text{X}$ 软件的汉化工作进行科研攻关。日前,国内已有成功的汉化 $\text{T}_{\text{E}}\text{X}$ 软件推向市场。在现有的各种版本的汉化 $\text{T}_{\text{E}}\text{X}$ 软件中,最杰出的当数CCT中西文排版系统。CCT系统是由中国科学院计算中心、中国科学院化学冶金研究所和中国科学院科技期刊编辑培训部联合开发的中西文激光照排系统。它是在 $\text{T}_{\text{E}}\text{X}$ 软件的基础上,保留了原软件(包括二次开发的排版软件,如 $\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 系统)的全部功能及特点,经汉化及功能扩展而成的。除了增加汉字的处理功能、实现中西文文字的混合排印之外,CCT照排系统还扩充了图形和图像的处理、坐标的转换、复杂表格的生成、化学结式的编排、汉字的自行造字和丰富的打印接口等功能。因而,CCT系统在中西文科技书刊排印方面与国内外同类型排版系统相比有独到之处。它的诞生为我国的科技工作者带来了福音,使得出版事业与国际接轨的速度大大加快了。

本书就是用CCT衍生的排版系统 $\text{C}\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 程序语言编排而成的。可以说,如果没有这样一个出色的汉化版本,本书的编写将是极其困难的。作者曾先后使用国内其它一些中文排版软件撰写初稿,但都无法达到现在 $\text{C}\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 系统的排版效果。

附录B将扼要介绍CCT系统的主要排版功能,并且简单叙述 $\text{C}\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 系统的排版命令、打印驱动程序和图像接口等使用方法。详细内容可阅读郭力等编著的CCT系统用户手册<sup>1</sup>。

## B.1 CCT系统的主要功能和特点

### 一、各种字型中西文的混合编排

CCT系统在保留 $\text{T}_{\text{E}}\text{X}$ 系统所具有的全部功能的同时,增加了对汉字的处理功能。系统可提供多种字号和字体的汉字,并实现了中西文的混合排版输出。例如对于适合微机系统的普及版(在 $\text{PCT}_{\text{E}}\text{X}$ 系统上扩展而成),CCT系统可提供 $24\times 24$ (适合24针打印机用)、 $64\times 64$ (适合激光打印机用)等点阵压缩字库和向量压缩字库。CCT系统提供的汉字库具有存贮空间小、字型精度高以及字体笔划边界光滑无毛疵等优点。

CCT系统的汉字字体共有四种:宋体、楷书体、黑体和仿宋体。与 $\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 系统一样,系统亦提供了十种标准字号的字体,不过用户也可根据需要自行设计特殊字号的大小。此外,如果需要打印罕见的系统字库未装入的汉字,用户还可自行造字。

在汉化的 $\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 排版系统( $\text{C}\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 系统)中,完全保留了原来西文字号和字体以及按英语语音拆字断行的排版功能,系统原有的各种排版控制命令(例如排版环境、图表命令、交叉引用等控制命令)在处理汉字时仍然有效。用户完全可以将汉字作常规的行文状态下的“字母”处理。熟悉 $\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 的用户只要在自己原来的硬件条件上再添加CCT软件,并在你的源文件(称之为 $\text{C}\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 文件)最前面的文件版式控制命令中调入汉化的文体类型宏定义文件(在原来的西文文体类型宏定义文件名前加一个“c”字即可,例如“carticle”、“cbook”),那么通过任何版本的中西文全屏幕文本编辑软件编制完源程序后,即可在执行排版后获得满意的中西文混合排版硬拷贝。

### 二、较强的图形绘制功能

从本书第八章中可以知道, $\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 系统只能画出一些简单的几何图形。由于图形功能比较弱,系统不能为用户自己提供的数据绘制比较复杂的曲线图形。而CCT系统则可以根据用户自备的数据(如实验测试数据、理论计算结果等)绘制各种曲线(不同粗细的虚线和实线),并能对用户提供的数据进行光滑处理。在图形中,CCT系统还可按照用户的要求自动绘制坐标轴、坐标值和网格等,并能在曲线或图形上加注各种标识符及中西文说明。

### 三、丰富的图像接口

CCT系统提供了和AutoCAD软件、GS等通用CAD软件以及图像扫描仪等的图像接口。如果用户从AutoCAD、GS等通用CAD软件,或者是从扫描仪中获得作为CCT

<sup>1</sup>郭力,张林波,葛向阳编著,CCT  $\text{T}_{\text{E}}\text{X}$ 软件中文接口:中外文科技激光照排系统用户手册,海洋出版社,北京,1993。



系统的图形或图像数据文件, 并且将这些文件参与源程序的编辑, 在排版输出时则可得到一次成功的图文并茂的精美拷贝。

#### 四、齐全的中西文DVI 文件驱动程序

在 $\text{\LaTeX}$ 系统中, 排版结果存在一个DVI 文件中。为了在屏幕上显示或在打印机上输出排版结果, 需要调用适当的设备驱动程序。使用CCT 系统时, 由于执行排版后生成的DVI 文件中含有汉字的特殊信息, 所以不能再简单调用 $\text{\LaTeX}$ 系统中原来配制的设备驱动程序来进行处理。为此, CCT 系统另外提供了功能齐全的适合不同类型显示器、打印机(差不多包括了国内所有的24 针和激光打印机) 的设备驱动程序。此外, 驱动程序允许用户通过事先准备好的含有图形或图像的数据文件调入系统后, 直接将生成的图形插入排版结果之中。

#### 五、新颖的化学结构式排版系统

$\text{\LaTeX}$ 系统中没有编排化学分子结构式的宏定义文件, 因此不支持分子结构式的排版。为了弥补这方面的不足, CCT 系统增加了一个交互式分子结构式的排版软件。用户可用它来绘制各种特殊的化学符号, 复杂的分子结构式, 脂肪环以及各种杂环等, 并且, 可将排出的分子结构式同正文一齐输出。化学结构式排版软件的绘图功能基本上可满足化工、冶金、生物和医学等领域科技书刊的排印要求。

#### 六、友善的软件环境

CCT 系统可以在MS-DOS 3.0 以上版本、各种中文DOS 版本或各类汉卡环境下正常运行。例如国内比较流行的金山Super-WPS 系统, 吴晓军213 CCDOS 系统, 联想汉卡等都能很好地支持CCT 系统。

## B.2 CCT 中西文排版系统的构成

CCT 系统主要由下列文件构成:

- 初始化程序: CCTINIT.EXE
- 预处理程序: CCT.EXE
- 屏幕显示驱动程序: DVISCR.EXE(适用于各种分辨率的显示适配器)
- 24 针打印机驱动程序: DVI24P.EXE(以及相应的驱动程序)
- 激光打印机驱动程序: DVILJP.EXE(适用于HP LaserJet 系列及兼容的激光打印机)
- 字号定义文件: CCT.DAT(用户可以自行定义特殊尺寸的各种字号的大小)
- $\text{\LaTeX}$  宏定义文件: C\*.STY(可处理汉字信息的各种文体类型文件)

- 绘图程序: PICTEX.EXE, PDFTOBF, INTFACE.STY
- 图像图形接口程序: PLTTODAT.EXE, TIFTOBMF.EXE

### B.3 CL<sub>A</sub>T<sub>E</sub>X 系统的基本流程

与T<sub>E</sub>X的衍生西文排版系统L<sub>A</sub>T<sub>E</sub>X一样,使用CCT系统的二次开发中西文排版软件CL<sub>A</sub>T<sub>E</sub>X之前,用户必须准备好自己的CL<sub>A</sub>T<sub>E</sub>X排版源文件。源文件的扩展名必须是“.CTX”。用户可以使用自己熟悉的任何一种中西文文本编辑系统来准备“CTX”文件。CL<sub>A</sub>T<sub>E</sub>X源文件中除了增加中文内容之外,绝大部分排版命令的格式与L<sub>A</sub>T<sub>E</sub>X源文件一样。由于L<sub>A</sub>T<sub>E</sub>X软件不能直接处理含有汉字的源文件,CCT系统提供了一个预处理程序CCT.EXE来将CL<sub>A</sub>T<sub>E</sub>X源文件转换成扩展名为.TEX的源文件。这样用户便可运行L<sub>A</sub>T<sub>E</sub>X软件来执行排版,从而产生可输出排版结果的DVI文件。因为在DVI文件中含有汉字信息,所以输出西文的专用驱动程序不再适用于显示或打印排版结果,所以必须使用CCT系统提供的新的驱动程序来输出排版结果。

归纳起来,使用CCT系统中的CL<sub>A</sub>T<sub>E</sub>X进行排版的主要过程包括以下几步:

- ① 编制CTX源文件。源文件中包含排版内容及相应的中西文排版命令。
- ② 运行CL<sub>A</sub>T<sub>E</sub>X批处理软件(其中包括(1)用CCT预处理程序CCT.EXE将CTX文件转换成TEX源文件;(2)运行L<sub>A</sub>T<sub>E</sub>X,对源文件执行排版。
- ③ 用CCT系统的驱动程序来显示或打印排版结果。

假定用户已经用中文文本编辑软件生成了一个CL<sub>A</sub>T<sub>E</sub>X标准源文件“sample.ctx”,如果要令CL<sub>A</sub>T<sub>E</sub>X为你执行排版工作,则可键入命令(假定用户已经进入了装有CL<sub>A</sub>T<sub>E</sub>X系统的PCTEX子目录下):

```
C:\PCTEX>clatex sample /
```

在CL<sub>A</sub>T<sub>E</sub>X运行过程中,系统首先对文件sample.ctx中的汉字进行预处理(将源文件sample.ctx转换为L<sub>A</sub>T<sub>E</sub>X读得懂的sample.tex文件),然后调用L<sub>A</sub>T<sub>E</sub>X对TEX源文件执行排版。排版结束后自动在PCTEX子目录生成了一个DVI文件。

如果运行顺利,用户想要显示排版结果,则可继续键入指令:

```
C:\PCTEX>cvview sample /
```

如果用户希望要在24针Epson LQ系列打印机上输出,那么可以在键盘上输入指令:

```
C:\PCTEX>dvi1q sample /
```

而若要在激光打印机上输出硬拷贝时,指令则变为:

```
C:\PCTEX>dvi1jp sample /
```

## B.4 CCT 初始化程序CCTINIT.EXE

软件CCTINIT.EXE 是用来初始化CCT 系统。它的作用是根据字号定义文件CCT.DAT 所定义的各个字号的大小生成 $\text{\LaTeX}$  系统排版时需要用到的TFM 文件(TeXFont Metric File) 以及宏定义文件CCHEAD.STY。在首次使用CCT 系统、或每次修改了CCT.DAT 文件中的某些参数之后必须先运行一次此程序, 从而确保整个系统的正常运行。

CCTINIT.EXE 的用法如下:

```
C:\PCTEX>cctinit (选择项)
```

其中用尖括号包含的内容表示可以省略的选择项部分(尖括号本身不用输入), 主要用来改变程序运行时的某些参数。每个选择项的第一个字符必须是“-”(减号), 后面紧跟一个字母(不区分大小写) 给出选择项的名称。有些选择项的后面还带有参数。注意符号“-”与选择项名及参数之间不能留有空格。不同的选择项之间必须用空格隔开。对于没有用选择项给出的参数, 初始化程序会自动选取一个缺省值。

例如下面指令:

```
C:\PCTEX>cctinit -s\hzfonts -x
```

中包含两个选择项“-s”和“-x”, 选择项“-s”带有参数“\hzfonts”。

CCTINIT.EXE 有如下一些选择项:

- s 后面必须跟随一个字符串参数, 其给出文件CCT.DAT 所在的路径名(子目录名)。缺省时为“\PCTEX”(适用于PCTEX)。
- t 后面必须跟随一个字符串参数, 从而给出放置程序产生的TFM 文件的路径名。缺省时为“\PCTEX\TEXFMS”(适用于PCTEX)。
- b 后面必须跟随一个字符串参数, 此选择项给出放置程序产生的“CCHEAD.STY”文件的路径名。缺省时为“\PCTEX\TEXINPUT” (适用于PCTEX)。
- x 没有参数。给出此选择项时CCT 利用 $\text{\LaTeX}$  的“\special” 命令来处理汉字的字体。而无此选择项时CCT 利用一个虚拟字库“CCDUMMY.TFM” 来处理汉字的字体。两种形式对于用户是等效的。

例如指令

```
C:\PCTEX>cctinit -hc:\tex\inputs -tc:\tex\fonts
```

指示CCTINIT.EXE 把生成的CCHEAD.STY 文件放在C: 盘的子目录“\TEX\INPUTS” 中, 而将生成的TFM 文件放在C: 盘的子目录“\TEX\FONTS” 中(这些子目录必须事先建立好)。

正常情况下用户不必给出任何选择项, 直接使用CCTINIT.EXE 设定的缺省值即可。

## B.5 CCT 预处理程序 CCT.EXE

软件CCT.EXE 可将CTX 文件或任意的带有汉字信息的源文件转换为 $\text{\LaTeX}$  读得懂的TEX 源文件。用户在准备好CTX 源文件后必须首先将其转化为TEX 文件, 然后才能进行排版处理。

调用CCT.EXE 的格式如下:

```
C:\PCTEX>cct [输入文件名[输出文件名]]
```

其中输入文件名为用户编制的CTX 源文件名, 输出文件名为转换得到的TEX 文件名。如果省略输入文件名中的扩展名, 那么系统会自动为你加上“.CTX” 作为输入文件名的扩展名; 如果省略输出文件名中的扩展名, 那么系统会自动为你加上“.TEX” 作为输出文件名的扩展名; 如果你完全省略输出文件名, 那么系统会认定输出文件的引用名与输入文件的引用名一样, 并另外加上“.TEX” 作为输出文件名的扩展名。例如命令

```
C:\PCTEX>cct sample
```

等价于

```
C:\PCTEX>cct sample.ctx sample.tex
```

CCT.EXE 除了产生一个TEX 文件外, 还会产生一个与 $\text{\LaTeX}$  源文件同名, 但扩展名为“.MAP” 的文件。这个文件中给出了 $\text{\LaTeX}$  源文件与TEX 文件中行号的对应关系, 以便用户在对 $\text{\LaTeX}$  源文件中查错时参考使用。“MAP” 文件中每行数字的排列具有如下形式:

$\text{\LaTeX}$  源文件中的行号  $\Rightarrow$  TEX 文件中的行号

例如, 某一CCT 文件生成的MAP 文件中包含下面一行:

```
5 ==> 6 7 8
```

则表示 $\text{\LaTeX}$  源文件中的第5 行对应着TEX 文件中的6, 7, 8 三行。

## B.6 字号定义文件 CCT.DAT

$\text{\LaTeX}$  系统的程序运行时, 总是先从CCT.DAT 文件中得到有关汉字大小、间距等信息。用户可以通过增添或修改此文件来设置各种中文字号的具体尺寸。如果修改了这个数据文件, 则需重新运行一次CCTINIT.EXE 程序, 以保证系统生成相应的TFM 文件及“CCHEAD.STY” 文件。

CCT.DAT 文件是一个普通的ASCII 文本文件。CCT 系统规定文件的第一行只有一个正整数, 它给出系统可以同时使用的不同尺寸(字体相同) 的字号个数。由于系统设置的允许出现的最大数值是26, 因此, 用户最多只能同时使用26 种不同大小的字号。在第一行以后, 须按顺序给出五列描述各种字号的数据。在每行数字中, 前面4 个都是正实

数或正整数, 而第5个数则必须是整数(可正可负)。这五列数字的具体含义如下:

- 第一列数: 字宽(以pt 为单位)
- 第二列数: 字高(以pt 为单位)
- 第三列数: 字距字宽(以pt 为单位) 与字宽之比
- 第四列数: 行距与字高之比
- 第五列数: 汉字字号控制命令中调用的参数

不同列或行的数据之间需用空格或回车(源文件换行标识符) 来隔开。

下面给出的是CCT 系统中配置的一种“CCT.DAT” 文件中的数据, 它定义了十种不同大小的标准汉字字号:

10				
9.0	9.0	0.05	0.3	-5
12.0	12.0	0.05	0.3	-4
36.0	36.0	0.05	0.3	0
27.5	27.5	0.05	0.3	1
21.0	21.0	0.05	0.3	2
16.0	16.0	0.05	0.3	3
13.75	13.75	0.05	0.3	4
10.5	10.5	0.05	0.3	5
8.0	8.0	0.05	0.3	6
5.25	5.25	0.05	0.3	7

例如倒数第三行数据定义的是通常书刊中的标准五号汉字(如本书正文中的基准汉字大小), 其大小为: 字宽=字高=10.5pt, 字距=10.5×0.05=0.525pt, 面行距=10.5×0.3=3.15pt。注意第五列整数中出现的负号表示汉字字号定义中的“小”。例如第二行中的“-5”表示“小五号”, 余此可类推。由于其中定义的字号属于汉字印刷字体中的标准字号, 用户最好不要修改它们的大小, 若要使用其它尺寸的汉字, 可以另行定义字号(可按照上述格式添在原始字号定义数据的下面, 注意所选字号参数不要与前面的参数发生冲突)。

## B.7 C<sub>l</sub>A<sub>T</sub>E<sub>X</sub> 专用排版命令

### B.7.1 C<sub>l</sub>A<sub>T</sub>E<sub>X</sub> 源文件的版式

C<sub>l</sub>A<sub>T</sub>E<sub>X</sub> 源文件与L<sub>A</sub>T<sub>E</sub>X源文件的编辑方式类似。正文(包括中文和英文等) 可以直接由键盘输入, 而指令仍然由英文字母和控制字符构成。用户可以用一个自己熟悉的中西文文本编辑程序来编辑C<sub>l</sub>A<sub>T</sub>E<sub>X</sub> 源文件。C<sub>l</sub>A<sub>T</sub>E<sub>X</sub> 排版源程序的扩展名应是“.CTX”。

为了使系统能处理汉字信息, 系统定义了一个宏文件“CCHEAD.STY”, 由于在C<sub>l</sub>A<sub>T</sub>E<sub>X</sub> 系统可选择的所有文体类型宏文件(以字母“C” 开头的“.STY” 文件) 已经调用了这个文件, 所以用户只须根据所要排版的文件类型在源文件开头设置文件版式命令时选择合适的参量(例如“carticle”(文章类)、“creport”(报告类)、“cbook”(书籍类)或“cletter”(信

表 B.1 汉字字体转换命令表

汉字字体	控制令	源程序实例	排版效果
宋体	<code>\songti</code>	<code>{\songti同济大学}</code>	同济大学
楷书体	<code>\kaishu</code>	<code>{\kaishu同济大学}</code>	同济大学
黑体	<code>\heiti</code>	<code>{\heiti同济大学}</code>	同济大学
仿宋体	<code>\fangsong</code>	<code>{\fangsong同济大学}</code>	同济大学

件类)等)。至于后续的源程序中排版命令,基本上就可按通常的 $\text{\LaTeX}$ 源文件那样来设置了。

### B.7.2 汉字字体转换命令

与 $\text{\LaTeX}$ 语言中的西文字体转换命令相类似,  $\text{\LaTeX}$ 系统中亦定义了“`\songti`”、“`\kaishu`”、“`\heiti`”和“`\fangsong`”四条命令来控制正文字体在宋体、楷书体、黑体和仿宋体之间转换。表B.1列举了四种汉字字体的控制令使用方式以及排版输出时的实际打印效果。

说明:

① 如果源程序中未对汉字字体设置任何转换命令,则系统将自动把基准字体设置为宋体。

② 对于局部使用特殊汉字字体的场合,可使用花括号分组控制符设置排版小环境,即汉字字体转换命令满足通常的 $\text{\LaTeX}$ 命令的作用范围规则。

③ 汉字字体的转换命令与西文字体的转换命令之间互相不受任何影响。

例如,源程序:

```
\it \fangsong 仿宋体命令对意大利斜体
''Italic type''\heiti\rm 无影响。 \songti
```

输出结果:

仿宋体命令对意大利斜体 “*Italic type*”无影响。

### B.7.3 汉字字号转换命令

在 $\text{\LaTeX}$ 系统中,汉字字号的转换方式有两种形式:一、使用与汉字匹配的 $\text{\LaTeX}$ 西文字号转换命令,实现中西文匹配字号的同步转换;二、使用汉字字号的专用转换命令,以打印特殊字型的汉字。

#### 一、使用与汉字匹配的 $\text{\LaTeX}$ 西文字号转换命令

考虑到 $\text{\LaTeX}$ 西文字号的转换命令使用比较方便,在 $\text{\LaTeX}$ 系统中,西文字号的

L<sub>A</sub>T<sub>E</sub>X 控制命令(如“\large”、“\small”等)已经被修改过(增加了同步转换汉字字号的功能)。所以用户在进行中西文混排时,只要直接调用本书表3.2中列出的西文字号的转换命令,就能非常容易地实现中西文匹配字号的同步转换。例如:

```
{\Large\heiti\bf Tongji University 同济大学},
{\large\it\fangsong Shanghai 上海}
```

可以得到:

**Tongji University 同济大学, Shanghai 上海**

注意,在L<sub>A</sub>T<sub>E</sub>X系统中使用西文字号转换命令时,原来设置的字体立即无效。L<sub>A</sub>T<sub>E</sub>X会将字体自动转换为罗马字体。如果需要特殊西文字体,则必须在字号转换命令后重新选择所需的字体。然而这个规则对于C<sub>1</sub>AT<sub>E</sub>X中的汉字字体转换命令作用范围毫无影响,即改变字号不会改变C<sub>1</sub>AT<sub>E</sub>X源文件前面所设定的字体。

## 二、使用汉字字号转换命令

除了借用西文L<sub>A</sub>T<sub>E</sub>X字号转换命令来同步改变汉字的大小之外,C<sub>1</sub>AT<sub>E</sub>X系统还为用户提供了一个专用的带单参数的汉字字号转换命令,其标准格式为:

```
\zihao{汉字字号参数}
```

其中汉字字号参数可选择系统配制的字号定义文件“CCT.DAT”中的第五列数字中的任意一个(参见第B.6节)。当用户需要自己扩充汉字字号时,字号参数选择的数目最多可以增加到26个。汉字字号转换命令的作用区域与通常的L<sub>A</sub>T<sub>E</sub>X西文字号命令一样,并同样遵守分组小环境排版规则。例如源程序

```
{\zihao{-4}\heiti 这是小四号黑体汉字},
{\fangsong\zihao{5} 这是仿宋体五号字},
{\songti\zihao{-5} 这是小五号宋体}
```

的输出结果为:

**这是小四号黑体汉字,这是仿宋体五号字,这是小五号宋体**

说明:

- ① 汉字字号转换命令“\zihao”不会改变前面设置的中文字体。
- ② 汉字字号转换命令“\zihao”仅对后续的中文有效。所以在所以使用汉字字号转换命令不会改变源程序原来设置的西文字号大小。

### B.7.4 汉字尺寸控制命令

在C<sub>1</sub>AT<sub>E</sub>X源程序中,除了可以使用标准的西文L<sub>A</sub>T<sub>E</sub>X中的标准长度单位之外,还可使用由如下三条命令定义的与汉字字号有关的相对单位作为尺寸控制命令中所需的长度值单位:

- `\ccwd` 当前字号的字宽与字距之和(width+inter word space)
- `\ccht` 当前字号的字高(height), 指一个字符在其所在行的基线以上部分的高度
- `\ccdp` 当前字号的字深(depth), 指一个字符在其所在行的基线以下部分的高度

例如在全局区域内输入缩排尺寸定义

```
\parindent = 2\ccwd
```

则可令每个中西文自然段开头缩进大约两个汉字宽度的空格。

在 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 源程序中, 用户不必去考虑如何调节每个字母或单词之间的距离。系统会考虑这方面的排版细节。鉴于中文字自身的特点(每个汉字的宽度相同, 排版时在一行中均匀散开, 可稀疏, 也可密集), 为了便于用户能够在 $\text{C}\text{I}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 源程序中修改字距, 系统定义了如下格式的汉字字距控制命令:

```
\ziju{相对字距}
```

其中参数相对字距 是用来设置后续行中当前字号的汉字间距与字宽之比。例如命令“`\ziju{0.35}`”可将当前字号的字距设定为字宽的0.35倍。

说明:

① 汉字字距控制命令仅对中文和西文单词之间的间距有效, 而对每个西文单词中的各字母之间的距离不发生作用。

② 汉字字距控制命令不受层次规则约束。一旦定义, 将对后续源文件的排版产生影响, 一直延续到下一个汉字字距控制命令出现为止。

### B.7.5 $\text{C}\text{I}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 与 西文 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的兼容性

尽管 $\text{C}\text{I}^{\text{A}}\text{T}_{\text{E}}\text{X}$  系统在处理汉字时保持了西文 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  原有的绝大部分功能, 但用户在编排中西文源文件时仍然要注意以下几点:

1. 汉字只能在行文或LR盒子中进行排版。如果要在数学公式状态中使用汉字, 必须使用LR盒子设置排版环境。例如可使用“`\mbox`”命令。此外, 排版过程中的数学公式上下标等自动缩小字号的规则在处理汉字时失效。用户在使用LR盒子设置的排版环境处理汉字时, 不要忘记根据具体场合选取合适的字号和字体。例如要想打印公式

$$F_{\text{下标}}^{\text{上标}}$$

则可输入源程序

```
\[
F^{\mbox{\scriptsize\heiti 上标}}_{\mbox{\scriptsize\songti 下标}}
\]
```



2. 同西文 $\text{\LaTeX}$ 一样,  $\text{\LaTeX}$ 系统会忽略所有多余的ASCII空格字符, 但汉字空格符(全角空格符, 占两个ASCII空格)不会被系统忽略。所以在使用CC-DOS的全角方式输入汉字时, 注意不要随意输入汉字空格符号。特别是在数学状态中, 全角空格会导致严重的语法错误。
3.  $\text{\LaTeX}$ 系统的源程序抄录命令“ $\backslash\text{verb}$ ”和抄录环境“ $\text{verbatim}$ ”不能用于含有汉字的部分。此命令在 $\text{\LaTeX}$ 系统中处理不包含汉字信息的西文文字部分时仍然有效。
4. 有些 $\text{\LaTeX}$ 命令会牵涉到浮动问题。如果命令所带的参量中另外含有控制命令(如汉字字体或字号命令), 则会给排版过程带来一些困难。例如章节标题的排版, 如果没有设置自动生成目录的控制命令, 则参量中含有设置标题的字体或字号命令不存在浮动问题。但是如果使用了列目录命令“ $\backslash\text{listofcontents}$ ”, 则在标题中使用汉字字型命令时须格外小心。例如本书中章次的标题字体与目录中出现的均选为黑体, 控制附录B标题的命令为

```
{\heiti \chapter{CCT 中西文排版系统简介}}
```

如果不想在目录中使用黑体(目录中的字体视前面中文的字体而定), 则可将上述程序改成

```
\chapter{\protect\heiti CCT 中西文排版系统简介}
```

在使用其它牵涉到浮动问题的控制命令时(如篇眉的处理, 图表注文等), 可使用类似的办法来处理。

## B.8 图形接口

$\text{\LaTeX}$ 允许用户直接将图形、图像插入所须排版的文件某个版面之中, 从而实现与文字部分的内容一起输出, 这样既可简化排版过程(省去了空出版面、粘贴图形两个步骤), 又做到了图文并茂。系统的图形、图像功能主要基于设备驱动程序所提供的图像接口。这个接口允许用户将符合一定格式的图像文件(称为BMF文件)插入到版面的任意位置上。所以, 原则上用户只需将自己的图形转换为BMF文件即可实现图文的同时输出。

### B.8.1 图形拼版命令

$\text{\TeX}$ 软件中提供了一条带单参量的特殊拼版控制命令 $\backslash\text{special}$ , 其使用格式为:

```
\special{拼版注解}
```

这个命令本身对排版的版面不会产生任何影响, 而只是将相应的控制命令及拼版注解拷贝到DVI文件的相应位置上。

CL<sup>A</sup>T<sub>E</sub>X利用这条命令来控制驱动器在当前位置插入一幅图形(已以标准数据文件形式存贮在磁盘之中), 具体格式为:

```
\special{BMF= 图形文件名}
```

其中的图形文件名 给出符合规定格式的黑白图像文件的文件名。根据系统规定, 图形文件的扩展名为“.BMF”, 所以图形文件通常亦称之为BMF 文件。驱动程序在DVI 文件中读到此命令时会自动将相应的BMF 文件所定义的图像拼在版面的当前位置上。例如图形拼版命令

```
\special{BMF=mypic.bmf}
```

可将用户自备的图形(由“mypic.bmf” 文件定义) 加进当前位置对应的版面上。

由于T<sub>E</sub>X在排版时拼版命令不作任何处理, 因此, 用户在利用它插入图像时应该在命令所在的版面上留出可容纳图像的足够空间。驱动程序在处理矩形图形拼版时以图形的左下角为原点, 即拼版命令所在的位置。因此, 用户可以使用横向空距命令“\hspace”和纵向(垂直)空距命令“\vspace”来调节图形拼版的合适空间。例如: 假定用户需要拼版的图形尺寸为长×宽=8cm×6cm, 已知版心宽度为14cm, 则可以采用如下居中插图的控制程序:

```
\vspace*{6cm}
\hspace*{3cm}
\special{BMF=mypic.bmf}
```

如果驱动程序在拼版时无法找到相应的BMF 图形文件, CL<sup>A</sup>T<sub>E</sub>X 将忽略拼版命令, 并给出一个警告信息。

### B.8.2 BMF 图形文件的生成及数据格式

如果用户手头拥有CCT 系统配置的图形处理程序, 则只要运行这些软件, 即可获得BMF 图形文件。例如, 利用一般扫描仪及图像处理软件, 可以将一些图片内容以标准图形数据形式存贮在TIF 格式文件之中。然后使用指令:

```
C:\PCTEX>TIFTOBMF <TIF 图形文件名>
```

则可将TIF 格式文件转换成引用名相同, 但扩展名为BMF 的图形文件。此外, 系统还自动生成了一个相同引用名的TEX 文件。其中包括了图形拼版命令和根据图形尺寸自动设置好参数的空距命令。这样, 用户只要将此文件插入CL<sup>A</sup>T<sub>E</sub>X 源程序中的适当位置, 就可以将图形加进正文之中。

BMF 文件中给出的是黑白图像的点阵数。文件开始是16 个字节的文件头, 它们依次给出:

- ① 横向分辨率 4 字节整数, 高位在前, 低位在后, 以 $2^{-16} \times \text{DPI}$ 为单位。
- ② 纵向分辨率 4 字节整数, 高位在前, 低位在后, 以 $2^{-16} \times \text{DPI}$ 为单位。

- ③ 图像宽度 双字节整数, 高位在前, 低位在后, 以点数为单位。
- ④ 图像高度 双字节整数, 高位在前, 低位在后, 以点数为单位。
- ⑤ 横向位移量 双字节整数, 高位在前, 低位在后, 以点数为单位。
- ⑥ 纵向位移量 双字节整数, 高位在前, 低位在后, 以点数为单位。

其中最后两个参数为带符号的整数, 取值范围为 $-32768 \sim 32767$ 。横向位移量表示应该将图像左移的点数, 而纵向位移量则表示应该将图像上移的点数。紧跟在文件头后面的是定义图像的点阵数据。图像中每行点从左到右每八个点分成一组(最后不足8点的也作为一组), 每组点用一个字节表示, 字节中的每个位元表示一个点(0 标识白点, 1 标识黑点)。八个点中最左边的对应字节中的最高位元。将代表所有点的字节按顺序排列, 即可得到图像的点阵数据。

作者曾与合作者一起使用上述格式, 成功地将金山中文排版系统的SPT图形编辑软件生成的图形数据文件转换成BMF文件, 由此可把SPT图形拼接到 $\text{C}\text{I}\text{A}\text{T}_{\text{E}}\text{X}$ 排版文件之中(参见图C.2)。

15A

## 附录 C

# 作图辅助软件 T<sub>E</sub>Xcad 简介

本附录介绍 L<sup>A</sup>T<sub>E</sub>X 排版系统的作图辅助软件 T<sub>E</sub>Xcad 的使用方法。

### C.1 引言

利用 L<sup>A</sup>T<sub>E</sub>X 程序设计语言中的绘图命令来绘制图形, 容易在源程序中对图形直接进行编辑或作局部修改, 这种方法比较受专业技术人员的欢迎。但是对于不太熟悉编程的 L<sup>A</sup>T<sub>E</sub>X 用户来说, 使用 L<sup>A</sup>T<sub>E</sub>X 系统的作图命令来绘制图形毕竟是一件不够直观的编辑工作。为了使 L<sup>A</sup>T<sub>E</sub>X 系统的作图过程能在图形状态下的显示屏幕上直观地进行, 一些计算机专家便致力于 L<sup>A</sup>T<sub>E</sub>X 系统辅助作图工具(CAD) 的开发、研制和完善工作。在各种样式的 L<sup>A</sup>T<sub>E</sub>X 排版系统的 CAD 软件之中, 德国计算机专家 Georg Horn 研制成功的与 L<sup>A</sup>T<sub>E</sub>X 绘图环境 picture 配套的作图辅助软件 T<sub>E</sub>Xcad 具有独特的风格。在最新版本的德英文 EMT<sub>E</sub>X 排版软件包中, T<sub>E</sub>Xcad 已经将作为一个重要的 L<sup>A</sup>T<sub>E</sub>X 排版系统向市场推出。

T<sub>E</sub>Xcad 是一个支持鼠标器的全屏幕图形编辑软件。通过鼠标器(或箭头位移键) 的简单操作, 用户可以非常方便地绘制一些常用的用 L<sup>A</sup>T<sub>E</sub>X 绘图语言定义的标准几何图形。这些图形编辑完毕之后, 系统输出的不是通常的图形数据文件(只能在特定的原驱动软件中重新显示), 而是自动生成一个含有一系列 L<sup>A</sup>T<sub>E</sub>X 绘图控制命令的 ASCII 码源程序。用户既可利用 T<sub>E</sub>Xcad 将其调入系统重新进行图形编辑或修改, 也可使用任意文本编辑软件直接对图形源程序作局部修改。编辑完毕之后, 用户可以将此文件插入自己的 L<sup>A</sup>T<sub>E</sub>X 源程序之中, 从而实现图文合并排版。不过, 比较好的排版模式是采用在主程序中插入命令“\input”的方法来将图形源程序调入系统。这样, 在需要修改的时候, 用户可以随时启动 T<sub>E</sub>Xcad 来将图形重新读入, 然后在屏幕上根据图形样式再作修改。

### C.2 T<sub>E</sub>Xcad 软件运行环境

T<sub>E</sub>Xcad 是用 Turbo Pascal 语言写成的全屏幕图形编辑软件, 能在所有 IBM PC/XT/AT 以及兼容机上运行。它既可以在西文 MS-DOS V3.0 及其以上版本的 DOS 环境中运行,

也可在国内流行的中文 CC-DOS 软件环境下运行。在使用时, 系统必须装有与图形显示器适配的 Turbo Pascal 图形驱动程序 BGI (Borland 图形接口)。例如用户可选择以下列出的一些 BGI 文件:

ATT.BGI	HERC.BGI	IBM8514.BGI
CGA.BGI	EGAVGA.BGI	PC3270.BGI

此外, 如果用户的计算机配有鼠标器, 那么在启动  $\text{T}_{\text{E}}\text{Xcad}$  之前, 应先将鼠标器驱动程序调入系统。

$\text{T}_{\text{E}}\text{Xcad}$  软件主要包括如下文件:

TEXCAD.EXE	$\text{T}_{\text{E}}\text{Xcad}$ 作图软件
TEXCAD.OPT	$\text{T}_{\text{E}}\text{Xcad}$ 作图系统选用参数
GH.BAT	含有选择菜单的批处理文件。用户可用来运行 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 、 $\text{T}_{\text{E}}\text{Xcad}$ 显示或打印驱动程序等。对于熟悉 MS-DOS 指令的用户来说, 可根据自己软件环境的具体情况修改对批处理文件作局部修改。
ASK.EXE	与 GH.BAT 配套的软件, 可由键盘选择用户需要运行的软件。
GH.TEX	用于调试由 $\text{T}_{\text{E}}\text{Xcad}$ 生成的作图源程序或其它 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 源程序。
*.BGI	与显示器适配的各种 Borland 图形接口驱动程序
EMLINES.STY	划线宏定义版式文件。定义可画任意斜率直线(简称为 EM 直线)的“\emline”控制命令。 <sup>1</sup>
EMLINES2.STY	新版本 $\text{E}_{\text{M}}\text{T}_{\text{E}}\text{X}$ 系统的划线宏定义文件。定义可画任意斜率直线的“\emlines”控制命令 <sup>2</sup> 。
TEXCAD.VER	过去所有 $\text{T}_{\text{E}}\text{Xcad}$ 版本的序号及简要记录。

安装  $\text{T}_{\text{E}}\text{Xcad}$  软件时, 可先将前面五个软件直接拷贝到用户自己的  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  子目录中, 并根据显示适配器分辨率选择合适的 Turbo Pascal BGI 文件。例如对于 VGA 显示适配器, 可选择拷贝“EGAVGA.BGI”文件至装有  $\text{T}_{\text{E}}\text{Xcad}$  软件的目录中。最后将“EMLINES.STY”和“EMLINES2.STY”两个文件复制到  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  宏定义文件库目录中(例如 \PCTEX\TEXINPUT)。

### C.3 $\text{T}_{\text{E}}\text{Xcad}$ 的启动和退出

对于计算机配置了鼠标器的场合, 在启动  $\text{T}_{\text{E}}\text{Xcad}$  之前, 用户应先运行鼠标器驱动程序(如果没有, 用户可直接运行  $\text{T}_{\text{E}}\text{Xcad}$ )。运行  $\text{T}_{\text{E}}\text{Xcad}$  的指令为:

C:\EMTEX>texcad . (图形文件名)

$\text{T}_{\text{E}}\text{Xcad}$  启动后首先与鼠标进行连接(如果失败, 会在屏幕显示警告信息), 然后在磁盘中寻找系统配置参数文件 TEXCAD.OPT, 其内容主要包括(左边一栏):

TeXcad 系统参数	省略参数
存取图形文件的子目录(Directories)	无(当前目录)
图形文件扩展名	.pic
剪取(存放) 和拼嵌(调入) 图像宏文件(Macro files) 子目录	无(当前目录)
宏文件扩展名	.mac
BGI 文件所在子目录	无(当前目录)
大十字光标快速移动切换开关(Snap)	off (关闭)
图像显示放大系数(Zoom)	1
标准L <sup>A</sup> T <sub>E</sub> X 线径单位尺寸(\unitlength)	1mm
画EM 直线控制开关(EMLines)	off (关闭)
EM 线径单位尺寸(EMLines/Linewidth)	0.4pt

当系统未读到TEXCAD.OPT 文件时, TeXcad 则会自动选取省略状态下的有关参数(用户可通过TeXcad 主菜单中的“Options”选择项自行修改“TEXCAD.OPT”文件中的参数, 并通过“Save Options”选择项将新的数据存到TEXCAD.OPT 文件中去)。在读完各种参数之后, 系统便调入BGI 图形接口驱动程序。如果找不到合适的BGI 程序, TeXcad 便在终端上向用户发出询问信息。

假如运行正常, 则终端屏幕会显示如图C.1 所示的作图窗口(左边)、信息窗口(右上)和选择菜单窗口(右下, 屏幕显示的是TeXcad 主菜单)。接下来, 用户便可使用鼠标器或箭头位移键来绘制或编辑文字和图形。

如果希望退出系统, 用户只须确认“Quit”项即可。不过, 如果用户忘记将当前编辑的文件存盘, 系统会在信息窗给出警告信息。

## C.4 菜单选择

在图C.1 中右下方菜单中的逆向光标(反视矩形亮条) 对应了可选择的图文编辑功能项。用户可移动鼠标器或连续掀按上下箭头键“↑, ↓”来控制移动光标(可选择项)。确认某一选择项时可按一下鼠标器左键或按回车键“Enter”。某些选择项被确认后还会引出下一级子菜单。在任何一级子菜单的顶部均用大写标记出菜单名称。图C.2 列出了TeXcad 主菜单和子菜单的完整结构图。如果要想从某一级子菜单中退出, 可以点按鼠标器右键或计算机键盘上的“Esc”键。

在TeXcad 菜单窗口主要显示如下几种常用的项类型:

- ① 子菜单引出下一级子菜单。
- ② 命令执行某一任务(图形的绘制、删除、存取等)。
- ③ 触发开关, 将某一特征参量置为On 或Off。
- ④ 通过信息窗口人机对话输入图形注解文字或数据以及系统所需的配置参数。

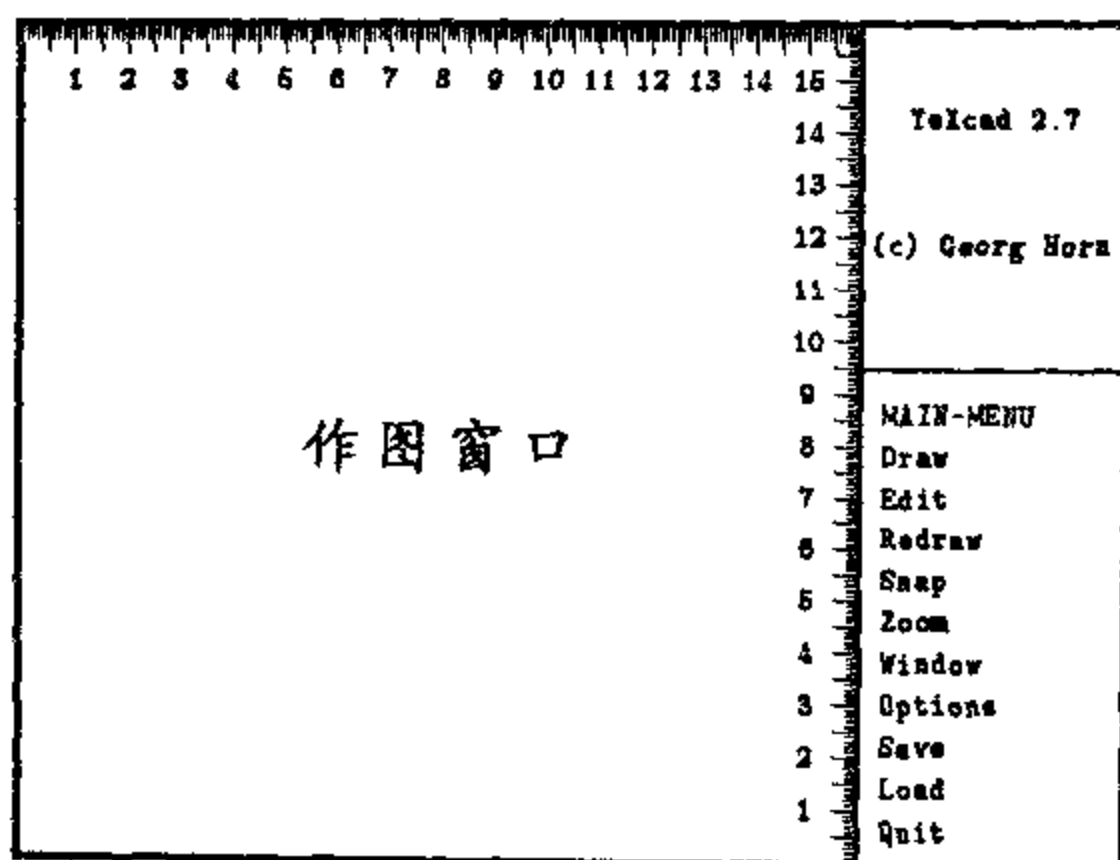


图 C.1 TeXcad 作图窗口示意图

## C.5 信息窗口

TeXcad 信息窗口的主要功能是在用户作图过程中给出提示信息。主要包括:

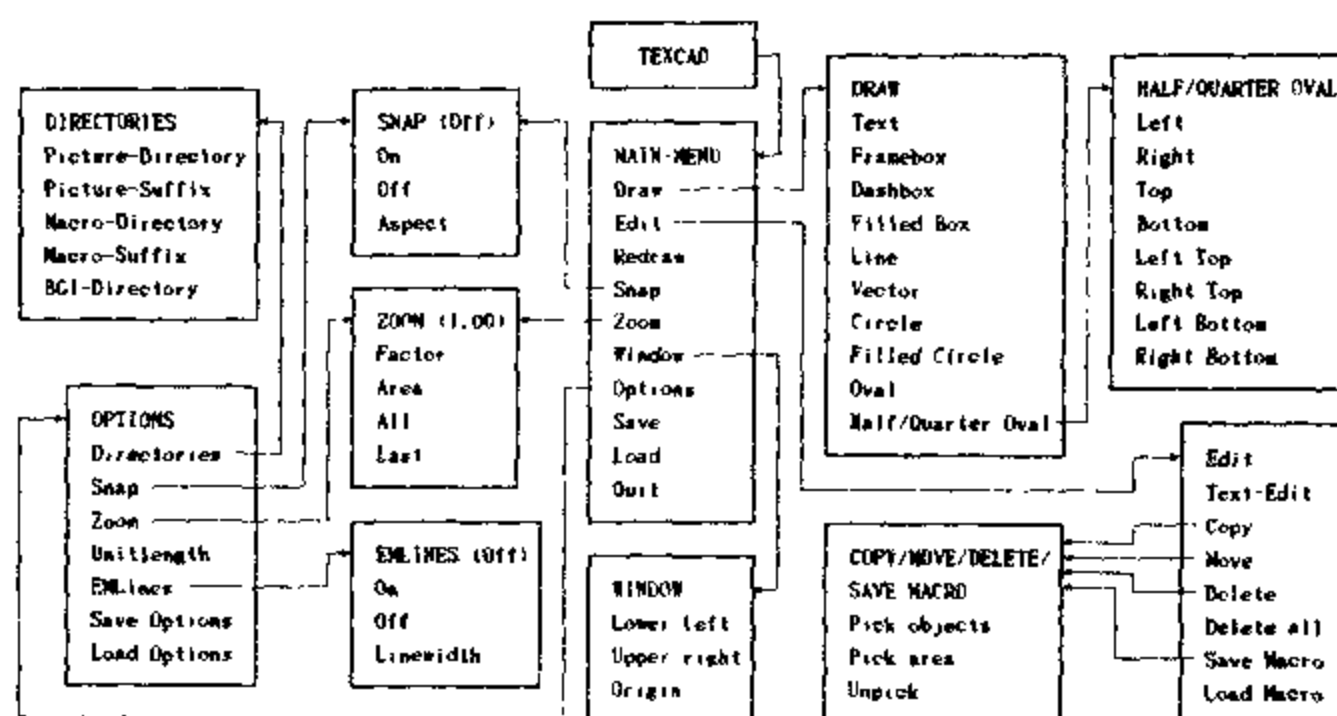
① 请求输入文字(例如文件名等) 用户可以通过按鼠标器的左右键或计算机键盘上的以下各键来帮助完成文字内容的输入。

- ←、→ 左右移动光标
- Ins(et) 在插入和覆盖两个状态中切换
- Del(ete) 清除光标处的字符
- Backspace(←) 清除光标左边的字符
- Esc 键或鼠标器右键 中断输入操作(并放弃输入的信息)
- Enter 键或鼠标器左键 结束输入, 并将文字信息发送给 TeXcad

② 请求选择作图参考点(例如矩形左下/右上角位置) 用户可使用“←、→、↑、↓”各键或拖拽鼠标器来移动作图窗口中的光标(十字线交叉点位置)。对于光鼠标器的用户, 则可以通过“Ctrl”键与箭头键的合并使用来加速十字线的移动。当选定位置后按鼠标器左键或回车键即可确认。如果用户发现选择有误, 可按鼠标器右键或 Esc 键放弃选择。

在操作过程中, 用户可使用 E、X、S 和 D 键来移动作图窗口, 并且可用 P 键来控制快速移动功能开关 Snap。



图 C.2 T<sub>E</sub>Xcad 主菜单和子菜单结构图

③ 显示错误信息 如果需要继续运行，可按回车键，或者按鼠标器左键。

④ Yes/No 询问 对于使用键盘的用户必须以“Y”或“N”回答，而对于使用鼠标器的用户则还可使用鼠标器左键或鼠标器右键来代替回答“Y”或“N”。

## C.6 使用 T<sub>E</sub>Xcad 窗口绘制图形

如果用户选择了作图项目(DRAW)，且在作图子菜单中又选择了所要绘制的某一类图形，那么当用户移动鼠标器或笔头键时，即可在图C.1所示的作图窗口看到顶角整个窗口的活动十字线。十字线的交叉点可用来选择作图的参考点，由十字线延长至边框的刻度值可确定坐标的相对数值。最小刻度值的实际长度可由选择项“Options/Unitlength”定义。例如作者在用T<sub>E</sub>Xcad制作图C.1时选用了等价于如下命令的参数：

```
\unitlength=0.90mm
```

### C.6.1 “粘贴”文字

作图菜单的第一个选择项是“Text”，可以用来在图形上添加西文图注文字。当你选定“粘贴”文字的参考点后(按鼠标器左键或Enter键完成操作，以下同)，作图窗口和信息窗口中的部分内容便会键系统暂时隐去，在作图窗口的上方留出一块等待用户输入文字的空间。这时，用户可按照通常编排LR排版盒子中内容的方式输入文字及字体或字号等控制命令。在结束文字输入操作之后，在菜单窗口会先后出现错向和纵向对位方式的选择菜单供用户选择文字相对参考点的对位方式。

对于错向对位方式，可供选择的项有：

- Left 文字排印时以参考点作为左边界，即文字内容将出现在参考点的右面。

- **right** 文字排印时以参考点作为右边界, 即文字内容将出现在参考点的左面。
- **center** 文字排印时将以参考点作为排版盒子的水平(横向) 对称中心。

对于纵向对位方式, 可供选择的项目有:

- **Top** 文字排印时以参考点作为上边界, 即文字内容将出现在参考点的下方。
- **bottom** 文字排印时以参考点作为下边界, 即文字内容将出现在参考点的上方。
- **center** 文字排印时将以参考点作为排版盒子的垂直(纵向) 对称中心。

当对位参量选定后, 系统将重新显示作图窗口中的图形(输入的文字用符号“T” 标识, 在将来排版输出时, 可得到文字信息), 并且回到作图子菜单, 等待用户输入下一个图形的绘制指令。

### C.6.2 实线、虚线和实心矩形盒子

选择“Framebox”、“Dashbox”或“Filled Box”可分别绘制实线、虚线或实心矩形盒子(对前面两种情形, 用户还可在盒子中填充文字)。

要绘制一个矩形盒子, 用户必须通过移动光标(十字线) 来分别选定矩形的左下角和右上角的参考点位置。而对虚线盒子, 用户还须在屏幕上任取两参考点, 以告诉系统你所希望的虚线中每一小段实线的长度。接下来, 用户便可在盒子中填写文字内容了(实心矩形盒子除外)。如果你只是想画一个矩形, 那么你必须得输入一个空格字符。输入文字的方式与上一小节基本相同, 只是在排印时, 对位参量控制文字版面与矩形框线相对位置的排版功能有所不同。

在版面的水平方向上:

- **Left** 文字排印时以矩形左边框线作为左边界, 即文字内容将在矩形盒子中左对齐排版。
- **right** 文字排印时以矩形右边框线作为右边界, 即文字内容将在矩形盒子中右对齐排版。
- **center** 文字排印时将在矩形盒子中水平居中排版。

而在版面的竖直方向上:

- **Top** 文字排印时以矩形顶线作为上边界, 即文字内容将出现在盒子内的顶部。
- **bottom** 文字排印时以矩形底线作为下边界, 即文字内容将出现在盒子内的底部。
- **center** 文字排印时将在矩形盒子中的竖直方向上居中排版。

每个盒子中的文字信息都将以符号“BT” 标识。

### C.6.3 直线与箭头

“Line”项和“Vector”项分别被用来画直线和绘制直线形状的箭头。用户只须利用十字线的移动先后设定起始点和终止点，就能获得直线或箭头。注意，系统会自动将上一根线段的终点设置为下一根线段的起始点。若要中断描画直线或箭头的操作，可按Esc 键或鼠标器右键来退出。

根据标准的L<sup>A</sup>T<sub>E</sub>X 作图规则，所画的直线或箭头只满足一些确定的斜率(参见第8.3.3和8.3.4 两小节)。换句话说，用户不能画任意斜率的直线或箭头。因此，在一般情况下T<sub>E</sub>Xcad 只能绘制具有特定斜率的直线或箭头，并力求接近用户所选取的联结两点的直线或箭头。

如果用户拥有可画任意斜率直线的E<sub>M</sub>T<sub>E</sub>X 驱动软件，那么，在画任意斜率直线之前，用户得先做以下两件准备工作：

① 在用户调用作图软件的L<sup>A</sup>T<sub>E</sub>X 源程序的文件格式命令中加进一条版式注解“emlines”(对高版本可使用“emlines2”)。

② 将主菜单中“Options”的子菜单中的“EMLlines”切换开关置于“On”(开启)状态。

这样，就可以画出任意斜率的直线了。在输出的含有排版命令的作图源程序将会多出一种命令

```
\emline{x1}{y1}{参考点1 序号}{x2}{y2}{参考点2 序号}
```

(系统会依次为用户选取的参考点逐一编上序号)。例如，作者在某次使用T<sub>E</sub>Xcad 随意绘制首尾相接的EM 直线时获得如下源程序：

```
\unitlength=1mm
\special{em:linewidth 0.4pt}
\linethickness{0.4pt}
\begin{picture}(114.33,117.33)
\emline{25.00}{24.33}{1}{93.00}{39.67}{2}
\emline{93.00}{39.67}{3}{113.67}{71.33}{4}
\emline{113.67}{71.33}{5}{79.00}{80.67}{6}
\emline{79.00}{80.67}{7}{114.33}{103.33}{8}
\emline{114.33}{103.33}{9}{66.67}{117.33}{10}
\end{picture}
```

当然，用户过去绘制的图形源程序仍可嵌套使用。

### C.6.4 圆和圆角矩形

空心圆和实心圆可分别通过选择“Circle”和“Filled Circle”项来绘制。操作时，参考点将作为圆心位置，稍微移动鼠标器(或箭头位移键)即可在屏幕上看到放大或缩小的圆。由于L<sup>A</sup>T<sub>E</sub>X 系统只允许绘制有限直径的圆，所以用户不要期望得到任意大小的

圆。在系统允许范围内选取某一尺寸的圆后, 用户可按鼠标器左键或Enter 键完成操作。

至于呈椭圆状的圆角矩形可由选择“Oval”项, 并采用与画圆相似的方法来绘制。而要画出圆角矩形的一半或四分之一图形则可选择“Half/Quarter Oval”项, 当然在此项选取后, 还有一个关于对位参量选取的子菜单需要用户选择(见图C.2)。

## C.7 图像编辑

主菜单的第二项是图像的编辑功能选择项“Edit”(见图C.2), 此选择项自身亦带一子菜单, 主要用于对已有的图形作局部或整体编辑与修改。例如图形中文字内容的更改、某个图形的复制、移动、擦除和存取等。

### C.7.1 图形定义

除了“Text-Edit”(文字修改)、“Delete all”和“Load Macro”之外, 其余四个选择项

Copy      Move      Delete      Save Macro

都须对所要修改的图形或区域作图形定义操作。在作图窗口中选择图形是通过这些项的子菜单中的选择项(Pick objects/ Pick area/Unpick) 来进行的:

- Pick objects 定义单个图形。当标识参考点的光标(十字线中心) 在鼠标器(或箭头位移键) 控制下按以下条件接近图形周围某个点时, 即可定义(激活) 该图形:

图形	参考点
文字	接近文字标识符T 的点
矩形盒子和圆角矩形	盒子内部的点
直线和箭头	接近直线上的点
圆	靠近圆周上的点

当完成图形选取操作后, 系统还会在信息显示窗口询问(图形是否选择正确):

Pick objects:

Pick [y/n]?

当用户按“Y”(或回车键、或鼠标器左键) 确认后, 该图形即被激活(以红线或虚线显示), 用户又可接着再选择定义其它图形; 如果不再需要定义任何图形, 则可按“Esc”键或鼠标器的右键返回。如果用户回答“N”, 系统则会在参考点周围另外再选取一个合适的图形供用户选择定义; 如果寻访失败, 信息窗口会显示有关提示信息。

- **Pick area** 定义含有若干个图形的矩形区域。此项操作时用户须利用鼠标器(或箭头位移键)选择左下角和右上角两个点的位置。当确定后,所有被包含在矩形框线中的图形都会被定义。
- **Unpick** 放弃定义。

### C.7.2 文字编辑

对图像中独立的文字内容(以“T”标记)或者是编排在矩形盒子中的文字(以“BT”标记)进行编辑的具体步骤比较简单:首先用户得在“Edit”子菜单下选择文字编辑功能项“Text-Edit”,然后控制鼠标器(或箭头位移键)来移动光标(十字线中心)接近需要编辑或修改的标识符“T”或“BT”位置,按鼠标器左键或“Enter”后,文字内容立即被定义。这时系统会在信息窗显示如下信息,以给用户一次确认的机会,

```
Pick Box/Text:
Edit [y/n]?
```

当用户确认后系统便将屏幕上图形全部隐去,在编辑窗口上方留出增添、修改或擦除文字的空间(被定义修改的文字内容会重现在屏幕上供用户修改)。用户可根据前面“粘贴”文字小节中介绍的文字输入方法来编辑文字,编辑结束后按鼠标器左键或“Enter”即可完或所定义的文字键辑工作(系统又返回到“Edit”子菜单中)。

### C.7.3 图像复制和移动

选择“Edit”子菜单的功能项“Copy”和“Move”可分别对被定义的单个图形或某个局部画面作复制和移动操作。

当用户定义并确认了需要进行复制或移动的图形或画面后,系统会要求你使用鼠标器(或箭头位移键)再选择两个参考点(表征将定义的图像复制或移到新位置的位移矢量)。确认后,对选择“Copy”情形,系统即将定义的图像复制到新的位置;而对选择“Move”的场合,图像被复制到新位置后原定义图形即被键除。

### C.7.4 图像删除

选择“Edit”子菜单中的功能项“Delete”可用于删除被定义的单个图形或某个局部画面。当用户定义了准备删除的图形或画面之后,系统会在信息窗中要求用户再次确认。

“Delete all”的功能是删除图形文件中全部的图像和文字内容。系统同样会在信息窗中要求用户予以再次确认。

### C.7.5 图形或画面的剪取和拼嵌

利用功能项“Save Macro”,用户可以将所编辑的图像中定义好的单个图形或小块画面以独立的完整 $\text{\LaTeX}$ 绘图源程序形式存放在磁盘中,用户既可以在需要时选择功能项

“Load Macro” 将其调入系统(拼嵌在某个画面之中) 也可将存放在磁盘中的图形源文件直接插入用户的排版主程序中。

在进行图形或画面的剪取和拼嵌过程中, 系统会请求用户键入存放或输出的文件名。文件名的选取可按照 MS-DOS 操作系统的规则。省略扩展名时, 系统会自动为用户取为“.mac”。在拼嵌画面操作过程中, 系统还会要求用户在屏幕上用光标给出标识输入画面位移矢量的两点位置。

## C.8 其它功能

除了绘图(Draw) 和编辑(Edit) 两大类功能选单之外, 系统还提供了许多有用的功能项, 可以帮助用户提高作图效率。

- **Redraw** 在主菜单中选择并确认此项后, 系统会将整个画面重新绘制一遍。在画圆或其它某些导致覆盖图形的场合, 使用此功能项可使画面刷新, 便于操作。
- **Snap** 选择此功能项子菜单中的“On”。后, 箭头位移键控制的绘图光标在移动时不再是缓慢的连续方式, 而是在画面上跳跃式的移动; 当选择“Off” 时快速移动功能失效。选择“Aspect” 项可改变大十字光标移动的步长(速度)。例如取数值“3” 可令光标每次移动三小格(单位长度的三倍, 见图C.1)。如果在作图过程中想切换快速移动开关, 可按“P” 键。
- **Zoom** 可以放大(改变标度) 在屏幕上显示的用矩形盒子定义的图像。在“Zoom” 的子菜单中有如下可选择的功能项:
  - **Factor** 选择图像放大倍数。
  - **Area** 整屏幕放大用矩形盒子定义的图像。
  - **All** 整版显示, 左下点设为坐标原点。
  - **Last** 显示在上一次标度下屏幕显示的图像。
- **Window** 此功能项可用于移动显示图像的模拟窗口。确认选择此项后, 用户应先在屏幕上设置一参考点, 然后在菜单中选择希望在窗口显示的图像方位。
  - **Lower left** 窗口移动后显示参考点右上方的图像。
  - **Upper right** 窗口移动后显示参考点左下方的图像。
  - **Origin** 原图形整版显示, 左下点设为坐标原点。
- **Options** 确认使用此功能项后, TeXcad 将允许用户根据此项引出的子菜单来更改“TEXCAD.OPT” 文件中的系统参数, 同时也提供了对正在编辑的图形调入或生成一个专用参数配置文件的途径。

## C.9 绘图源程序的调入和生成

在T<sub>E</sub>Xcad 主菜单最后一项“Quit”上面两项是“Load”和“Save”。选用“Load”项可让用户调入IAT<sub>E</sub>X 绘图源程序；选用“Save”项可将用户绘图过程中生成的源程序存盘。使用这两个功能项时，系统将请求用户键入调入或生成的文件路径名和文件名。

当使用“Load”时，系统会给出警告信息：

```
Picture not saved!
Load anyway [y/n]?
```

与拼嵌图像文件不同，调用功能项“Load”会冲掉(删除)当前编排的全部图形。如果用户施行调入图形文件操作时键入MS-DOS 通配符“\*”或“?”，那么系统会在菜单窗口向用户提供有关图形文件的目录。用户在移动反视光标时对需要编辑的文件予以确认即可完成调入操作。

使用“Save”输出的任何IAT<sub>E</sub>X 绘图源程序都将是准确无误的规范程序，通常具有如下形式：

```
\unitlength=1.00mm
\special{em:linewidth 0.4pt}
\linethickness{0.4pt}
\begin{picture}(76.47,39.61)
\put(28.46,1.94){\circle{3.89}}
\put(2.13,23.61){\circle*{4.27}}
\put(51.46,39.61){\oval(14.00,0.00)[]}
\put(53.80,12.94){\line(5,4){22.67}}
\put(50.14,33.27){\line(3,-2){25.67}}
.....
\end{picture}
```

当调用“EMLines”选择项时，系统会将相应的画直线命令改成\emline 命令。

如果用户需要用文本编辑软件对绘图源程序作修改，请注意以下几点：

① 对T<sub>E</sub>Xcad 来说，在调入修改后的绘图源程序时，会忽略掉在绘图环境(picture) 以外出现的所有控制命令。因此在绘图环境(picture) 以外添加的所有控制命令对主程序也许有用，但对T<sub>E</sub>Xcad 毫无意义。

② 除了以“\put”或“\emline”形式构成的绘图命令之外，T<sub>E</sub>Xcad 在调入修改后的绘图源程序后，不会将对出现在绘图环境(picture) 之中的其它排版控制命令作任何变动。

③ 如果用户希望在主程序中将图形居中或按其它特殊对位方式排版，可以在“\begin{picture}(L<sub>x</sub>, L<sub>y</sub>)”命令后面另外添加坐标原点参数(x<sub>0</sub>, y<sub>0</sub>) 来调节图形在版面上的合适位置(参见第8.2 节)。





## 附录 D

# 出错信息

### D.1 L<sup>A</sup>T<sub>E</sub>X 系统出错信息

以下按字母顺序列出了所有L<sup>A</sup>T<sub>E</sub>X排版系统的出错信息以及导致错误的可能因素。注意, 屏幕显示的L<sup>A</sup>T<sub>E</sub>X错误信息均以下面两行提示文字

LaTeX error. See LaTeX manual for explanation.

Type H <return> for immediate help.

来先行引出的。面T<sub>E</sub>X系统出错信息则没有这两行文字。

- ! Bad \line or \vector argument.

绘图模式中的画直线注解命令“\line”或“\vector”中确定斜率的一对参数 $(x, y)$ 非法。“\line”命令中参数的定义域为区域 $[-6, 6]$ 间的整数, “\vector”命令中参数的定义域为区域 $[-4, 4]$ 间的整数, 同时要选择两参数的比值 $y/x$ 为最简分数(参见8.3.3或8.3.4小节)。

- ! Bad math environment delimiter.

系统在数学排版状态中发现了无法匹配的再次进入数学公式状态的模式切换控制符“\[”或“\ (“; 或是在LR排版盒子中, 或是在行文排版状态之中发现了退出数学公式状态的控制符“\]”或“\)”。

- ! Bad use of \.

在段与段之间引进了多余的毫无意义的换行控制符“\.”。这个错误发生在对位排版环境命令(如行居中center、行左齐flushleft和行右齐flushright环境)或是对位控制命令(如行居中\centering、行左齐raggeleft和行右齐raggeright控制命令)中(参见4.4和4.5节)。

- \begin{西文环境注解名1} ended by \end{西文环境注解名2}.

环境起始命令与结束命令的环境注解名不匹配。出现这种语法错误的主要原因为:

①用户可能在输入西文环境注解名发生了拼写错误; ②使用环境排版命令时可能

输入多余的起始命令“\begin”; ③程序员遗忘了一个结束命令“\end”; ④嵌套使用环境命令时同一层次的各自配对的环境命令之间发生交迭。

- ! can be used only in preamble.

在文稿排版起始命令“\begin{document}”以后误用了只能在全局说明区域(在命令“\begin{document}”之前)使用的控制命令(例如“\documentstyle”、“\includeonly”和“\makeindex”等)。如果源文件中出现了一个多余的“\begin{document}”, 也会出现上述错误。

- ! Command name ... already used.

用户在使用命令\newcommand、\newenvironment、\newlength、\newsavebox 或\newtheorem 自定义某种功能的排版控制命令或排版环境时, 误用了系统保留的或者是已经定义了的控制命令名。解决这个问题最简单的办法是换一个名字。如果要想改变现有的排版命令的功能, 应使用命令“\renewcommand”或“\renewenvironment”。

- ! Counter too large.

在使用英文字母作为序码数字字体时, 序号大于26。出现这种错误很可能是用户正在罗列条目很多的表, 或是用户自己重新设置了计数器的数值。

- ! Eenvironment ... undefined.

用户误用了未经定义的非排版环境控制命令。

- ! Float(s) lost.

用户在小页排版盒子“\parbox”、小页排版环境(minipage)、脚注命令“\footnote”或浮动图表等盒子中嵌套使用了插图环境(figure)、表格(table)环境, 或旁注命令“\marginpar”。这是一个输出错误, 排版时会有一个或多个插形、表格或旁注发生丢失现象。

- ! Illegal character in array arg.

在矩阵元排版环境、或者制表环境的列参量中、或是在表格列扩充命令“\multicolumn”中的第二个参量中有非法字符。

- ! Missing \begin{document}.

在需要排版的正文文稿之前未设置文稿排版环境起始命令“\begin{document}”。产生此语法错误除了可能遗忘“\begin{document}”之外, 还可能在全局说明区域内含有错误的控制命令, 比如丢失转码控制符“\”或是忘了输入位于花括号内的控制参量。

- ! Missing p-arg in array arg.

在矩阵元排版环境、或者制表环境的列参量中、或是在表格列扩充命令“\multicolumn”的第二个参量中所使用的“p”注解缺少用花括号括出的宽度说明参数。

- ! Missing  $\theta$ -exp in array arg.

在矩阵元排版环境、或者制表环境的列参量中、或是在表格列扩充命令“\multicolumn”的第二个参量中所使用的“ $\theta$ ”注解缺少用“ $\theta$ -”引出的表达式。

- ! No such counter.

在命令“\setcounter”或“\addtocounter”中调用了不存在的计数器。这种错误很可能是由于拼写错误引起的。然而，如果这个错误信息是系统在读AUX文件时产生的，那么可能是程序员在非全局说明区域使用了“\newcounter”命令。

- ! Not in outer par mode.

用户在数学排版状态下或是在广义的小页排版盒子(parbox)中使用了插图环境(figure)、表格(table)环境，或旁注命令“\marginpar”。

- ! \pushtabs and \poptabs don't match.

L<sup>A</sup>T<sub>E</sub>X 发现了一个没有“\pushtabs”与之匹配的“\poptabs”命令，或者在系统读到命令“\end{tabbing}”之前仍有一个或多个尚未匹配的“\poptabs”。

- ! Something's wrong--perhaps a missing \item.

导致这个错误的原因可能是在条目列表环境中丢失了一个条目控制命令“\item”，或者是在参考文献排版环境(thebibliography)中没有给出环境参数说明。

- ! Tab overflow.

控制符“\=”的个数超出了L<sup>A</sup>T<sub>E</sub>X系统在tabbing环境中所允许的最大个数。

- ! There's no line here to end.

在自然段之间使用了毫无意义的换行命令“\newline”或换行控制符“\\”。如果用户想要在自然段之间的纵向方向上额外空出一些间距，可以直接使用纵向空距命令“\vspace”

- ! This may be a \LaTeX bug.

L<sup>A</sup>T<sub>E</sub>X 系统无法解释当前的排版错误，这可能是由于前面的错误带来的；如果屏幕显示的错误是执行排版过程中出现的第一个错误，则有可能是L<sup>A</sup>T<sub>E</sub>X系统本身有错误(这时，可与L<sup>A</sup>T<sub>E</sub>X系统经销商联系)。

- ! Too deeply nested.

用户在条目列表环境中嵌套了太多层次的子条目列表环境。最多可以嵌套的层数取决于用户所拥有的L<sup>A</sup>T<sub>E</sub>X版本。但一般来说，标准的情形是至少可以嵌套四层。这对大多数情形已经是足够用了。

- ! Too many unprocessed floats.

排版源文件中包含了太多的旁注命令“\marginpar”。此外，如果可浮动的图表太多而超过了L<sup>A</sup>T<sub>E</sub>X排版系统的内存空间，则亦会带来这个错误。解决这个问题的方法之一是在编制源文件时尽可能将可浮动的环境远离文件结束的位置。

- ! Undefined tab position.

试图在采用定位控制符“\=”定义过的位置上使用控制符“\>”、“\+”、“\=”或“\<”。

- ! in mid line.

试图在“\tabbing”环境中的一行中间使用控制符“\<”，控制符“\<”只能用于一行的开头。

## D.2 TeX 系统出错信息

本节按字母顺序列出了常见的TeX系统的出错信息以及引起错误的原因。

- ! Counter too large.

当脚注序号用专用字符(如\*, †, ‡等)打印时, 如果序号超过了9就会产生这种错误。此外, 如果使用了太多的“\thanks”也会引起这个错误。

- ! Double subscript.

在编排下标(脚标)中嵌套下标时, 忘了添加低层次分组控制符号, 例如当编排数学式子“ $A_2$ ”时输入了有语法错误的源程序“ $\$A_{2}_{3}\$$ ”。正确的排版程序应是“ $\$A_{2_{3}}\$$ ”。

- ! Double superscript.

在编排上标中继续嵌套上标时, 忘了添加低层次分组控制符号, 例如当编排数学式子“ $A^2$ ”时输入了有语法错误的源程序“ $\$A^{2}^{3}\$$ ”。正确的排版程序应是“ $\$A^{2^{3}}\$$ ”。

- ! Extra alignment tab has been changed by \cr.

在矩阵排版环境中或是在制表环境中的某一行内输入了超过定义太多的列项, 即在一行中出现了多余的隔开列元素的控制符“\$”。产生这种错误的原因大多数是由于忘记在前面输入的一行元素末尾添上换行号“\”。

- ! Extra } or forgotten \$.

分组符号“{ }”或切换数学排版状态的控制令(例如“\$”)不匹配。用户可能在源程序中忘记输入“{”、“\{”、“\{”或“\$”。

- ! Font ... not loaded: Not enough room left.

源文件调入了过多的超过系统有效内存所能容纳的新字库。解决问题的办法是将源文件分成若干个子文件再进行排版。

- ! I can't find file '...'. .

系统无法在磁盘中找到用户在源程序中指定读入的文件。这很可能是用户在使用命令“\input”或“\include”时, 将所需调入的文件名拼写错了; 或者是在文件格式命令“\documentstyle”将文体类型参量写错了。当系统发现此错误时会暂时停止运行, 并在屏幕上显示:

Please type another input file name:

以给用户重新输入文件名的机会。用户可在冒号“:”后面输入正确的文件名后按回车键即可动态地改正此错误。

- ! Illegal parameter number in definition of ...

用户在使用`\newcommand`、`\renewcommand`、`\newenvironment`或`\renewenvironment`定义命令的源程序中, 设置虚拟参量时出现控制符“#”语法错误。充当虚拟参量的控制符“#”在输入时后面应跟一个正整数。

- ! Illegal unit of measure (Opt inserted).

如果用户刚遇到一个

! Missing number, treated as zero.

错误, 那么这是前面语法错误中的一部分; 否则, 意味着 $\text{\TeX}$ 系统发现了一个无单位的长度参数(例如将应该为0.5 cm的参数仅写成一个数字0.5)。如果用户按回车键继续, 则系统将自动以点(pt)为单位。有些尺度控制命令在却省长度参数时也会出现这种错误。

- ! Misplaced alignment tab character &.

字符“&”是 $\text{\TeX}$ 系统保留的特殊控制符, 只能用在矩阵排版环境或者制表环境之中。如果在其它环境中单独出现“&”则会产生此错误。如果用户需要打印符号&, 应使用控制符“\&”。

- ! Missing control sequence inserted.

在自定义命令`\newcommand`、`\renewcommand`、`\newlength`或`\newsavebox`中, 第一个参量不是一个正确的控制命令表达式。

- ! Missing number, treated as zero.

在需要长度作为参量的尺度控制命令中丢失了参数, 或者只写了单位而未输入数字, 或者在具有可选择项的命令中忘记了方括号“]”。此外, 若将控制命令“`\protect`”置于一个尺度控制命令或者可以产生数字的命令(如“`\value`”)之前也会产生这类错误。

- ! Missing { inserted.

! Missing } inserted.

$\text{\TeX}$ 系统发现控制符“{”或“}”不匹配。注意系统发出错误信息的当前位置不一定是真正有语法错误的位置。

- ! Missing \$ inserted.

系统在行文状态或排版盒子中发现了只能在数学排版状态下才能使用的数学公式控制令。如果在数学排版模式中出现了一个空行亦会导致此错误。

- ! Not a letter.

在指示系统如何拆字的控制命令中出现了参量输入错误。

- ! Paragraph ended before ... was complete.

在不允许空行作为参量的排版命令中出现了一个空行。用户很可能在源程序中某个命令的结束位置忘记了一个右花括号"}。

- ! \scriptfont ... is undefined (character ...).

! \scriptscriptfont ... is undefined (character ...).

! \textfont ... is undefined (character ...).

在数学公式排版状态下使用了不常使用的字库(系统预先未调入)。例如用小体大写字母"\sc"作为数学公式的下标。解决问题的方法是事先使用调字库命令"\load", 例如:

```
\load{\footnotesize}{\sc}
```

就可将脚注大小的小体大写字母调入系统。

- ! TeX capacity exceeded, sorry [...].

TeX 系统的某个内存空间溢出, 从而导致系统停止运行。在正常情况下, 特别是对于只有几页内容的小文件, 这种错误很少发生。因此, 用户可以将源文件分解成几个小文件来重新执行排版, 如果这类问题仍然存在, 则可确定源文件中有错误; 否则, 就有可能计算机可运行的基本内存不够<sup>1</sup>。在错误信息的后面, 系统会给出内存溢出的类型, 下面列出的是一些最常见的部分:

- buffer size: 章节命令(如"\section")、图表注文命令"\caption"等命令中输入的参量(例如字符串)太长。尽管系统在执行列目录"\tableofcontents"或其它命令时, 此类错误已经发生, 但是出错信息只会在系统使读到文稿排版环境结束命令"\end{document}"时才会在屏幕显示出来。用户可以采取将上述命令中的参量缩短一些来避免这种错误发生。
- exception dictionary: 用户在给系统指示拆字方式时, 使用了过多的"\hyphenation"命令。解决问题的办法是减少一些为不常用的单词作拆字方式注解的控制命令, 多采用控制符"\-"来解决拆字与断行问题。
- hash size: 用户文件自定义的命令过多或交叉引用的标号太多。
- input stack size: 这个错误通常是发生在自定义命令之中。例如利用一个需要定义的命令来定义这个命令本身:

```
\newcommand{\fwg}{and \fwg}
```

会导致死循环, 直到内存溢出为止。

<sup>1</sup>对于用CC-DOS启动的系统, 由于显示汉字字库占据了较大的内存, 运行L<sup>A</sup>T<sub>E</sub>X便会出现内存空间溢出错误; 对于计算机基本内存被病毒感染系统, 执行排版时亦会出现类似错误。

- **main memory size:** TeX 系统的主内存溢出, 发生这类错误的原因可能有三种: (1) 定义了过多的太长、太复杂的命令; (2) 在一页内使用了过多的“\index”或“\glossary”; (3) 当前页版面的内容太复杂, 以至于主内存空间里容纳不了这页的所有信息。解决前两个问题的办法比较简单, 用户可以尽量少用“\index”或“\glossary”。但是第三种原因导致的错误就比较难解决。错误的真正的原因可能是信息量过大的“tabbing”、“tabular”、“array”或“picture”环境。此外, 过多的浮动图表也会导致此类错误。确定错误原因的方法之一是在出错位置插入命令“\clearpage”, 如果错误消失, 则证明确实是系统主内存耗尽, 否则的话, 说明源程序有错误。对于主内存溢出问题, 可在发生错误的那一页源程序内插入换页命令“\newpage”; 涉及浮动图表导致的错误, 可以将其改成非浮动的, 或尽量将浮动图表远离文件的结尾位置。
  - **poolsize:** 用户使用了过多的交叉引用标号或是自行定义了过多的新命令。另外, 在命令“\newenvironment”、“\newtheorem”以及与计数器有关的控制命令(如“\setcounter”)中丢失参量最后的花括号亦会导致此类错误。
  - **save size:** 源程序中嵌套使用控制命令、环境命令或自定义命令的层数太多。
- **! Text line contains an invalid character.**  
在用户使用其它文字编辑软件编制源文件时, 留下了怪异字符(例如其它软件的排版控制符)。
  - **! Undefined control sequence.**  
系统遇到了非法的控制命令, 用户可能在输入命令时拼写有误。此外, 某些命令放错位置(在条目列表环境外面使用“\item”命令), 或者遗忘文件格式命令“\documentstyle”亦会产生此类错误。
  - **! Use of ... doesn't match its definition.**  
假如“...”是个LaTeX命令, 那么它可能是个作图命令, 在它所携带的参量中有一个语法错误; 如果是“\@array”命令同其定义不符, 那么在“array”或“tabular”环境参量中的“\@-”表达式有错误。
  - **! You can't use macro parameter character #' in ... mode.**  
在行文排版状态中使用了系统保留字符“#”, 这很可能是用户想打印这个字符。解决的办法是在字符“#”前补上控制符“\”。





# 附录 E

## 常用 L<sup>A</sup>T<sub>E</sub>X 命令

### 句子和段落

引号 单引号'...' 双引号''...''  
 破折号 连接字母- 连接数字-- 破折号---  
 字距 窄字距\, 词间距\, 字间距\, 句尾标识符\@.  
 特殊字符 \\$\@ &\@ %\@ #\@ {\@ }  
 \) -\\_  
 强调 {\em...}  
 盒子 \mbox{...}  
 脚注 \footnote{...}  
 日期 \today

### 字 体

\rm	Rom	罗马字体	\it	Ital	意大利体
\bf	Bold	黑体	\sl	Slan	倾斜体
\sf	SSrf	等线体	\sc	CAPS	小体大写
\tt	Type	打字体	\mit	Mit	数学字体
	\songti	宋体		\kaishu	楷书
	\heiti	黑体		\fangsong	仿宋

### 字 号

\Huge	\LARGE	\normalsize	\footnotesize
\huge	\Large	\small	\scriptsize
	\large		\tiny

\sihsa{n} (C<sub>L</sub>A<sub>T</sub>E<sub>X</sub>命令, 对西文无效)

### 特征符号修饰

$\dot{v}$	\dot{v}	$\bar{v}$	\bar{v}	$\check{v}$	\check{v}	$\hat{v}$	\hat{v}
$\acute{v}$	\acute{v}	$\grave{v}$	\grave{v}	$\eth{v}$	\eth{v}	$\v{v}$	\v{v}
$\circ{v}$	\circ{v}	$\lrcorner{v}$	\lrcorner{v}	$\eth{vv}$	\eth{vv}	$\underline{v}$	\underline{v}
$\breve{v}$	\breve{v}	$\u{v}$	\u{v}				

### 章节划分和目录

\part	\section	\paragraph
\chapter	\subsection	\subparagraph
	\subsubsection	

\appendix 进入附录版式  
 \tableofcontents 生成章节目录  
 \listoffigures 生成图形目录  
 \listoftables 生成表格目录

### 数学公式

$\$...\$$  或  $\{(...)\}$  行中公式  
 $$$...$$$  或  $[...]$  独立显示公式  
 \begin{equation}...\end{equation} 编号方程  
 \begin{eqnarray}...\end{eqnarray} 多行编号方程,  
 类似三列状矩阵; 命令\nonumber省略某一行  
 方程式的编号; 带星号命令形式(eqnarray\*)  
 则省略多行编号方程环境中的所有编号。

\_{...} 下标

^{...} 上标

' 撇号(')

\frac{分子}{分母} 打印分式  $\frac{分子}{分母}$

\sqrt[n]{arg} 打印根式  $\sqrt[n]{arg}$

省略号 \ldots \cdots \vdots \ddots

数学符号 见表6.3-6.10

希腊字母 见表6.2

界标 由命令\left或\right引出(见表6.13)

$\overline{\exp}$  打印 $\overline{\exp}$

“带帽”见表6.14

间距 见表6.1

### 独立显示行文

$\begin{quote} \dots \end{quote}$  编排小型引文

$\begin{quotation} \dots \end{quotation}$  编排较长的引文

$\begin{center} \dots \end{center}$  编排多行居中行文, 行由 $\backslash$ 分开

$\begin{verse} \dots \end{verse}$  编排诗文, 诗行间由 $\backslash$ 分开

$\begin{verbatim} \dots \end{verbatim}$  程序抄录状态

### 条目列表

$\begin{itemize} \dots \end{itemize}$  编排行首打印标记的条目

$\begin{enumerate} \dots \end{enumerate}$  编号条目

$\begin{description} \dots \end{description}$  贴以描述型标签的条目

### 文体和页眉版式

$\documentstyle[版式注解]{文体类型}$

版式注解:

11pt twoaside leqno

12pt twocolumn fleqn

titlepage

revtex preprint(APS 版式注解)

版式类型:

article report book letter

carticle creport cbook cletter( $\text{\LaTeX}$ )

aps(APS 版式)

$\pagestyle{页眉版式}$  页眉版式:

plain empty headings myheadings

$\pagenumbering{页码参数}$  页码参数:

arabic roman alph Roman Alph

### 题名页

$\maketitle$  根据控制命令

$\title \backslash author \backslash date$ (可选任意形式)

生成题名部分版面

$\begin{titlepage} \dots \end{titlepage}$  自行设计题名页版面

$\begin{abstract} \dots \end{abstract}$  编排摘要

### 交叉引用

$\label{key}$  设置引用标识符 $key$

$\ref{key}$  打印标识符 $key$ 所含值

$\pageref{key}$  打印标识符 $key$ 所处位置的页码值

### 文献引用

$\begin{thebibliography}{lbl} \dots \end{thebibliography}$  编排参考文献目录;  $lbl$  为最长的序号字符串; 每条文献由下面一行命令给出

$\bibitem[lbl]{key}$  (用序号 $lbl$ ) 编排一条由关键词 $key$ 标识的文献

$\cite[note]{keys}$  引用由关键词 $key$ 标识的文献 (用 $note$ 作为附加注文)

### 源文件分解输入

$\input{file}$  排版时调入文件 $file$

$\include{file}$  排版时调入文件 $file$  (在下面一行命令中拒绝的文件除外)

$\includeonly{files}$  排版时仅允许调入在 $files$ 中注名的文件(文件名之间用逗号隔开)

### 换行

$\newline$  无条件换行

$\linebreak[n]$  强迫断行;  $0 \leq n \leq 4$

$\nolinebreak[n]$  不准断行;  $0 \leq n \leq 4$

$\ll[len]$  另起一行, 行距为 $len$

$\-$  可拆字(断字)位置

$\begin{sloppypar} \dots \end{sloppypar}$  为自然段设置断字与断行环境

$\sloppy$  allow loose lines

### 换页

$\pagebreak[n]$  强迫换页;  $0 \leq n \leq 4$

$\nopagebreak[n]$  不准换页;  $0 \leq n \leq 4$

$\samepage$  只能在段落之间换页

$\newpage$  另面排版

$\clearpage$  打印所有图表, 并另起一页

### 命子

$\mbox{...}$

$\makebox[wd][pos]{...}$   $wd$ : 盒子宽度;  $pos$  为位置参数: 1(靠左); 1(靠右); 省略(居中)

`\fbox{text}`  
`\framebox[wd][pos]{text}` 功能与 `\mbox` 或 `\makebox` 相同, 但边框线紧靠正文边界  
`\newsavebox{cmd}` 定义盒子或图形库命令 `cmd`  
`\sbox{cmd}{text}`  
`\savebox{cmd}[wd][pos]{text}` 排版功能与 `\mbox` 或 `\makebox` 相同, 但输出至库 `cmd` 中  
`\usebox{cmd}` 打印库 `cmd` 中存储的盒子  
`\begin{minipage}[pos]{wd}` 小页正文  
`\end{minipage}` 小页环境(宽度: `wd`, 对位参量: `t`(顶部); `b`(底部); 省略(中心线))  
`\parbox[pos]{wd}{小页正文}` 同小页环境, 但没有脚注

### 尺寸控制

长度单位: `cm mm in pc pt em ex`  
`\ccwd \ccht \ccdp`

`\newlength{cmd}` 定义长度命令 `cmd`  
`\setlength{cmd}{len}` 为长度命令 `cmd` 赋值 `len`  
`\addlength{cmd}{len}` 为命令 `cmd` 增加长度值 `len`  
`\settowidth{cmd}{txt}` 命令 `cmd` 的长度值取为行文 `txt` 的宽度

### 空距

`\hspace{len}` 横向空距; 带星号形式功能更强, 行首亦有效  
`\vspace{len}` 纵向空距; 带星号形式功能更强, 页面顶部同样有效  
`\hfill` 横向撑满  
`\hrulefill` 用下划线横向撑满  
`\dotfill` 用点横向撑满  
`\vfill` 纵向撑满

### 作图

`\begin{picture}(L_x, L_y)(x, y)` 作图命令  
`\end{picture}`  
 图形长  $\times$  宽  $= L_x \times L_y$ ;  
 左下角坐标值  $(x, y)$   
`\put(x, y){...}` 图形放在点  $(x, y)$   
`\multiput(x, y)(\Delta x, \Delta y){n}{...}` 从点  $(x, y)$  起以增量  $(\Delta x, \Delta y)$  复制  $n$  个图形

`\makebox(L_x, L_y)[pos]{...}` 产生  $L_x \times L_y$  盒子。  
 图中文字位置参量 `pos`: `t`(顶部); `b`(底部); `l`(靠左); `r`(靠右); 省略(居中)。命令 `\framebox` 和 `\savebox` 给出类似图形。  
`\dashbox{d}(L_x, L_y)[pos]{...}` 功能与 `\makebox` 相似, 但盒子用虚线(尺度为  $d$ ) 框出。  
`\line(\Delta x, \Delta y)(L_x)` 斜率为  $\Delta y/\Delta x$  且水平方向长度为  $L_x$  的直线(如果  $\Delta x = 0$  则线长亦为  $L_x$ ),  $0 \leq |\Delta x|, |\Delta y| \leq 6$   
`\vector(\Delta x, \Delta y)(L_x)` 与直线命令 `\line` 相似, 但添加了箭头记号, 并且  $0 \leq |\Delta x|, |\Delta y| \leq 4$ 。  
`\shortstack[pos]{...}` 类似表格排版环境 "tabular"  
`\circle{d}` 直径为  $d$  的空心圆;  
 - 形式为实心圆。  
`\oval(x, y)[part]`  $x \times y$  圆角矩形的一部分。  
`\frame{...}` 环绕图形的方框  
`\thinlines` 细线  
`\thicklines` 粗线  
`\unitlength` 图形基本单位的实际长度

### 图表环境

`\begin{figure}...\end{figure}` 活动插图  
`\begin{table}...\end{table}` 活动表格  
`\caption{...}` 图表注文

### 表格命令

行由 " \ " 分开; 列由下列控制字符决定:  
`\ =` 设置位标  
`\ >` 移至下一个位标  
`\kill` 当前行不输出

### 矩阵和表格环境

`\begin{array}[pos]{cols}...\end{array}`  
`\begin{tabular}[pos]{cols}...\end{tabular}` 环境  
`array` 用于编排矩阵, `tabular` 用于编排表格;  
 每个元素用 `&` 隔开, 而每行元素用 `\\` 隔开;  
 对位参量 `pos`: `t`(顶部), `b`(底部), 省略(居中)  
 列参量 `cols`: `l`(左对齐); `r`(右对齐); 省略(居中);  
`|` 竖线  
`p{len}` 设置列的宽度为 `len` `{...}` 列与列之间的文字或空距  
`*{n}{...}` 等价于项目  $(...)$  的  $n$  重拷贝

$\backslash\text{multicolumn}\{n\}\{col\}\{\dots\}$  本列项目宽度设置为下一行中的 $n$ 列宽度, 列参量为 $col$

$\backslash\text{hline}$  在行元素之间划水平线

$\backslash\text{cline}\{i-j\}$  在 $i-j$ 列之间划水平线

### 自定义命令

$\backslash\text{newcommand}\{cmd\}[n]\{\dots\}$  定义带 $n$ 个参量的控制命令 $cmd$

$\backslash\text{newenvironment}\{nam\}[n]\{beg\}\{end\}$  定义带 $n$ 个参量的环境 $nam$

$\backslash\text{newtheorem}\{nam\}\{cap\}$  定义一个定理式(计数方式为 $cap$ )排版环境

### 计数器

$\backslash\text{setcounter}\{ctr\}\{n\}$  为计数器赋值 $n$

$\backslash\text{addtocounter}\{ctr\}\{\Delta n\}$  为计数器添加数值 $\Delta n$

# 参考文献

- [1] Arthur L. Samuel, *First Grade T<sub>E</sub>X: A Beginner's T<sub>E</sub>X Manual*, Stanford University, 1983.
- [2] Donald E. Knuth, *The T<sub>E</sub>Xbook*. Addison-Wesley, Reading, Massachusetts, 1984.
- [3] Michael Urban, *An Introduction to L<sup>A</sup>T<sub>E</sub>X*, T<sub>E</sub>X Users Group, Providence, USA, 1986.
- [4] Leslie Lamport, *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*, Addison-Wesley, Reading, Massachusetts, 1985.
- [5] 郭力, 张林波, 葛向阳编著, *CCT (T<sub>E</sub>X 软件中文接口): 中外文科技激光照排系统用户手册*, 海洋出版社, 北京, 1993.
- [6] 张福炎、蒋新儿、李滨宇编著, *微型计算机IBM PC 的原理与应用*, 南京大学出版社, 南京, 1990.
- [7] 江建名编著, *著编译审技指南*, 中国科学技术大学出版社, 合肥, 1988.
- [8] 唐兴汉、张元林编著, *编辑出版印刷手册*, 科学普及出版社, 北京, 1987.
- [9] 陈伟光、刘敏钰、王国超、王战军编写, *微机排版系统使用大全*, 陕西电子编辑部, 西安, 1992.
- [10] 丁卫星、赖天树编著, *L<sup>A</sup>T<sub>E</sub>X 排版软件实用教程*, 中国科学技术大学出版社, 合肥, 1993.



# 索引

□ (空格), 12, 91  
 \\_, 91  
 ! (感叹号), 13, 198  
 \!, 25, 91  
 !' (i), 44  
 ", 14  
 \\*{...}, 44  
 #, 15, 157  
 \# (#), 15  
 \$, 15, 28  
 \$...\$, 88  
 \\$ (\$), 15  
 \$\$...\$\$, 89  
 %, 15  
 \% (%), 15  
 &, 15, 103, 120  
 \& (&), 15  
 ' (单上引号), 13, 14  
 '' (双上引号), 14  
 \', 44, 118  
 ' (单下引号), 13  
 '' (双下引号), 14  
 \', 44, 118  
 \'{...}, 25  
 ( (左括号), 13, 96, 106  
 \[, 88  
 \(...\), 88  
 ) (右括号), 13, 96, 106  
 \], 88  
 \* (星号), 13, 123  
 \*- 表达式, 123  
 + (+), 13, 14, 96  
 \+, 118  
 - (连字号-; 减号-), 13, 14, 96, 118, 204

\-, 13, 204  
 -- (-), 14  
 --- (—), 14  
 . (西文句号, 小数点), 13  
 \., 44  
 / (除号), 13, 96, 106  
 \/, 28  
 : (冒号), 13  
 \:, 91  
 ; (分号), 13  
 \;, 91  
 . (逗号), 13  
 \, , 25, 91  
 {, 15  
 {} (分组控制符), 26, 32  
 \{ ( { ), 15, 106  
 }, 15  
 \} (}), 15, 106  
 < (小于号), 88, 96  
 \<, 118  
 = (等号), 88, 96  
 \=, 17, 44, 117  
 > (大于号), 88, 96  
 \>, 117  
 ? (问号), 44  
 ?' (¿), 13  
 @, 13, 123  
 \@, 13  
 [ (左方括号), 13, 96, 106  
 \[, 26  
 \[...], 89  
 ] (右方括号), 13, 106  
 \], 26  
 \ (反斜杠转码符), 9, 25, 91

- $\backslash$  (换行), 12, 103, 117  
 $\backslash[ \dots ]$ , 35  
 $\sim$ , 15  
 $\sim\{ \dots \}$ , 92  
 $\backslash\sim$ , 15, 44  
 $\backslash\sim\{ \dots \}$ , 15, 25  
 $\backslash\sim\{ \}$ , 15  
 $\_$  (下划线), 15  
 $\_ \{ \dots \}$ , 93  
 $\_ \_$ , 15  
 $|$  (竖线), 106, 120  
 $||$  (双竖线), 120  
 $| |$  (||), 97  
 $\sim$ , 15  
 $\sim$ , 44  
 $\backslash\sim\{ \}$ , 15  
 $\backslash a^*$ , 118  
 $\backslash a'$ , 118  
 $\backslash a=$ , 118  
 $\backslash aa$  (ä), 44  
 $\backslash AA$  (Å), 44  
 $\backslash acute$ , 112  
 $\backslash address$ , 184  
 $\backslash addtocounter$ , 169  
 $\backslash addtolength$ , 37  
 $\backslash as$  (æ), 44  
 $\backslash AE$  (≥), 44  
 $\backslash aleph$  (ℵ), 97  
 $\backslash alph$ , 170  
 $\backslash Alph$ , 170  
 $\backslash alpha$  (α), 95  
 $\backslash amalg$  (II), 96  
 $\backslash angle$  (∠), 97  
 $\backslash appendix$ , 72  
 $\backslash approx$  (≈), 96  
 $\backslash arabic$ , 169  
 $\backslash arccos$  (arccos), 100  
 $\backslash arcsin$  (arcsin), 100  
 $\backslash arctan$  (arctan), 100  
 $\backslash arg$  (arg), 100  
 $\backslash ast$  (\*), 96  
 $\backslash asymp$  (∞), 96  
 $\backslash author$ , 74; (APS 版式), 189  
 $b$  (底线齐), 76, 105, 116, 125, 138  
 $\backslash b$ , 44  
 $\backslash backslash$  (\), 15, 97, 106  
 $\backslash bar$ , 112  
 $\backslash baselineskip$ , 83  
 $\backslash baselinestretch$ , 82, 188  
 $\backslash begin\{abstract\} \dots \backslash end\{abstract\}$ , 78;  
 (APS 版式), 190  
 $\backslash begin\{array\} \dots \backslash end\{array\}$ , 103, 104  
 $\backslash begin\{center\} \dots \backslash end\{center\}$ , 52  
 $\backslash begin\{description\} \dots \backslash end\{description\}$ , 62  
 $\backslash begin\{displaymath\} \dots \backslash end\{displaymath\}$ , 89  
 $\backslash begin\{document\}$ , 11, 48  
 $\backslash begin\{document\} \dots \backslash end\{document\}$ , 48  
 $\backslash begin\{enumerate\} \dots \backslash end\{enumerate\}$ , 60  
 $\backslash begin\{eqnarray\} \dots \backslash end\{eqnarray\}$ , 108  
 $\backslash begin\{eqnarray*\} \dots \backslash end\{eqnarray*\}$ , 109  
 $\backslash begin\{equation\} \dots \backslash end\{equation\}$ , 90  
 $\backslash begin\{figure\} \dots \backslash end\{figure\}$ , 125  
 $\backslash begin\{flushleft\} \dots \backslash end\{flushleft\}$ , 55  
 $\backslash begin\{flushright\} \dots \backslash end\{flushright\}$ , 54  
 $\backslash begin\{instit\} \dots \backslash end\{instit\}$   
 (APS 版式专用命令), 190  
 $\backslash begin\{itemize\} \dots \backslash end\{itemize\}$ , 61  
 $\backslash begin\{letter\} \dots \backslash end\{letter\}$ , 185  
 $\backslash begin\{list\} \dots \backslash end\{list\}$ , 174  
 $\backslash begin\{math\} \dots \backslash end\{math\}$ , 88  
 $\backslash begin\{minipage\} \dots \backslash end\{minipage\}$ , 76  
 $\backslash begin\{picture\} \dots \backslash end\{picture\}$ , 127  
 $\backslash begin\{quotation\} \dots \backslash end\{quotation\}$ , 47, 49  
 $\backslash begin\{quote\} \dots \backslash end\{quote\}$ , 49  
 $\backslash begin\{references\} \dots \backslash end\{references\}$   
 (APS 版式专用命令), 192  
 $\backslash begin\{sloppypar\} \dots \backslash end\{sloppypar\}$ , 180  
 $\backslash begin\{tabbing\} \dots \backslash end\{tabbing\}$ , 117  
 $\backslash begin\{table\} \dots \backslash end\{table\}$ , 115  
 $\backslash begin\{tabular\} \dots \backslash end\{tabular\}$ , 120  
 $\backslash begin\{thebibliography\} \dots$   
 $\backslash end\{thebibliography\}$ , 148  
 $\backslash begin\{theindex\} \dots \backslash end\{theindex\}$ , 152  
 $\backslash begin\{titlepage\} \dots \backslash end\{titlepage\}$ , 75  
 $\backslash begin\{title\} \dots \backslash end\{title\}$   
 (APS 版式专用命令), 189



- `\begin{verbatim}... \end{verbatim}`, 58  
`\begin{verbatim*}... \end{verbatim*}`, 59  
`\begin{verse}... \end{verse}`, 51  
`\begin{字体说明词}...`  
`\end{字体说明词}`, 28  
`\beta` ( $\beta$ ), 95  
`\bf` (黑体), 27  
`\bibitem`, 148  
`\bigcap` ( $\cap$ ), 98  
`\bigcirc` ( $\bigcirc$ ), 96  
`\bigcup` ( $\cup$ ), 98  
`\bigodot` ( $\odot$ ), 98  
`\bigoplus` ( $\oplus$ ), 98  
`\bigotimes` ( $\otimes$ ), 98  
`\bigkip`, 36  
`\bigskipamount`, 36  
`\bigsqcup` ( $\sqcup$ ), 98  
`\bigtriangledown` ( $\nabla$ ), 96  
`\bigtriangleup` ( $\triangle$ ), 96  
`\biguplus` ( $\uplus$ ), 98  
`\bigvee` ( $\vee$ ), 98  
`\bigwedge` ( $\wedge$ ), 98  
`\bmod`, 102  
`\Box` ( $\square$ ), 97  
`\bot` ( $\perp$ ), 97  
`\bowtie` ( $\bowtie$ ), 96  
`\breve`, 112  
`\bullet` ( $\bullet$ ), 61, 96  
`c` (居中), 103, 122  
`\c`, 44  
`\cal` (数学花体), 95  
`\cap` ( $\cap$ ), 96  
`\caption`, 116, 126  
`\case` (分式; APS 版式专用命令), 195  
`\cc`, 186  
`\ccwd` (CCT 系统专用命令), 218  
`\ccwh` (CCT 系统专用命令), 218  
`\ccwh` (CCT 系统专用命令), 218  
`\cdot` ( $\cdot$ ), 96  
`\cdots` ( $\cdots$ ), 14, 99  
`\centering`, 57  
`\centerline`, 53  
`\chapter`, 66  
`\check`, 112  
`\chi` ( $\chi$ ), 95  
`\circ` ( $\circ$ ), 96  
`\circle`, 136  
`\circle*`, 137  
`\cite`, 149  
`\cleardoublepage`, 13, 67  
`\clearpage`, 13, 67  
`\cline`, 121  
`\closing`, 185  
`\clubsuit` ( $\clubsuit$ ), 97  
`cm`, 34  
`\columnsep`, 84  
`\columnseprule`, 84  
`\cong` ( $\cong$ ), 96  
`\coprod` ( $\prod$ ), 98  
`\copyright` ( $\copyright$ ), 43  
`\cos` ( $\cos$ ), 100  
`\cosh` ( $\cosh$ ), 100  
`\cot` ( $\cot$ ), 100  
`\coth` ( $\coth$ ), 100  
`\csc` ( $\csc$ ), 100  
`\cup` ( $\cup$ ), 96  
`\d`, 44  
`\deg` ( $^\circ$ ), 43  
`\dagger` ( $\dagger$ ), 96  
`\dashbox`, 130  
`\dashv` ( $\dashv$ ), 96  
`\date`, 74  
`\ddag` ( $\ddagger$ ), 43  
`\ddagger` ( $\ddagger$ ), 96  
`\ddot`, 112  
`\ddots` ( $\ddots$ ), 99  
`\def` (TeX 命令), 188  
`\deg` ( $\deg$ ), 100  
`\delta` ( $\delta$ ), 95  
`\Delta` ( $\Delta$ ), 95  
`\det` ( $\det$ ), 100  
`\diamond` ( $\diamond$ ), 96  
`\Diamond` ( $\Diamond$ ), 97  
`\diamondsuit` ( $\diamondsuit$ ), 97  
`\dim` ( $\dim$ ), 100

- \div ( $\div$ ), 96
- \documentstyle, 9
- \dot, 112
- \dotfill, 85
- \doteq ( $\doteq$ ), 96
- \downarrow ( $\downarrow$ ), 97, 106
- \Downarrow ( $\Downarrow$ ), 97, 106
- \draft (APS 版式专用命令), 188
- \ell ( $\ell$ ), 97
- em, 33
- \em (强调字体), 28
- \emline (EMTeX 系统专用命令), 224
- \emptyset ( $\emptyset$ ), 97
- \encl, 186
- \end{document}, 11
- \epsilon ( $\epsilon$ ), 95
- \eqnum (APS 版式专用命令), 194
- \equiv ( $\equiv$ ), 96
- \eta ( $\eta$ ), 95
- \evensidemargin, 81
- \ex, 33
- \exists ( $\exists$ ), 97
- \exp (exp), 100
- \fangsong (汉字仿宋体, CCT 系统专用命令), 216
- \fbox, 39, 129
- \figure (APS 版式专用命令), 193
- \fill, 84, 85
- \FL (APS 版式专用命令), 195
- \flat ( $\flat$ ), 97
- \fnsymbol, 170
- \footheight, 82
- \footnote, 67, 170
- \footnotemark, 68
- \footnotesize, 30
- \footnotetext, 68
- \footskip, 82
- \forall ( $\forall$ ), 97
- \frac, 101
- \framebox, 39, 128
- \frown ( $\frown$ ), 96
- \gamma ( $\gamma$ ), 95
- \Gamma ( $\Gamma$ ), 95
- \gcd (gcd), 100
- \ge ( $\geq$ ), 96
- \geq ( $\geq$ ), 96
- \gg ( $\gg$ ), 96
- \glossary, 153
- \grave, 112
- h, 116, 125
- \H, 44
- \hat, 112
- \hbar ( $\hbar$ ), 97
- \headheight, 82
- \headsep, 82
- \heartsuit ( $\heartsuit$ ), 97
- \heiti (汉字黑体, CCT 系统专用命令), 216
- \hrfill, 85
- \hline, 120
- \hom (hom), 100
- \hookleftarrow ( $\hookleftarrow$ ), 97
- \hookrightarrow ( $\hookrightarrow$ ), 97
- \hrulefill, 85
- \hspace, 34
- \hspace\*, 34
- \huge, 30
- \Huge, 30
- \hyphenation, 179, 205
- \i ( $i$ ), 25
- \Im ( $\Im$ ), 97
- \inath ( $\imath$ ), 97, 113
- in, 34
- \in ( $\in$ ), 96
- \include, 167
- \includeonly, 167
- \indent, 84
- \index, 127, 151
- \indexentry, 151
- \indexspace, 152
- \inf (inf), 100
- \infty ( $\infty$ ), 97
- \input, 166
- \int ( $\int$ ), 98
- \iota ( $\iota$ ), 95
- \it (意大利斜体), 27

- `\item`, 60, 152  
`\itemindent`, 174  
`\itemsep`, 174  
`\j` ( $j$ ), 25  
`\jmath` ( $j$ ), 97, 113  
`\Join`, 96  
`\kaishu` (汉字楷书, CCT 系统专用命令), 216  
`\kappa` ( $\kappa$ ), 95  
`\ker` ( $\ker$ ), 100  
`\kill`, 118  
`l` (左齐), 38, 103, 120, 122, 129, 138  
`\l`, 44  
`\L` ( $L$ ), 44  
`\label`, 145  
`\labelsep`, 174  
`\labelwidth`, 175  
`\lambda` ( $\lambda$ ), 95  
`\Lambda` ( $\Lambda$ ), 95  
`\langle` ( $\langle$ ), 106  
`\large`, 30  
`\Large` ( $\bigg$ ), 30  
`\LARGE`, 30  
`\LaTeX` ( $\text{\LaTeX}$ ), 9, 16  
`\lceil` ( $\lceil$ ), 106  
`\ldots` ( $\dots$ ), 14, 99  
`\le` ( $\leq$ ), 96  
`\leadsto` ( $\leadsto$ ), 97  
`\left`, 106  
`\leftarrow` ( $\leftarrow$ ), 97  
`\Leftrightarrow` ( $\Leftrightarrow$ ), 97  
`\leftharpoondown` ( $\leftharpoondown$ ), 97  
`\leftharpoonup` ( $\leftharpoonup$ ), 97  
`\leftline`, 55  
`\leftmargin`, 174  
`\leftrightarrow` ( $\leftrightarrow$ ), 97  
`\Leftrightarrow` ( $\Leftrightarrow$ ), 97  
`\leq` ( $\leq$ ), 96  
`\lfloor` ( $\lfloor$ ), 106  
`\lg` ( $\lg$ ), 100  
`\lhd` ( $\lhd$ ), 96  
`\lim` ( $\lim$ ), 100  
`\liminf` ( $\liminf$ ), 100  
`\limsup` ( $\limsup$ ), 100  
`\line`, 132  
`\linebreak`, 179, 205  
`\linebreak[n]`, 180  
`\linethickness`, 127  
`\listoffigures`, 126, 144  
`\listoftables`, 116, 144  
`\listparindent`, 174  
`\ll` ( $\ll$ ), 97  
`\ln` ( $\ln$ ), 100  
`\load`, 31  
`\location`, 184  
`\log` ( $\log$ ), 100  
`\longleftarrow` ( $\longleftarrow$ ), 97  
`\Longleftarrow` ( $\Longleftarrow$ ), 97  
`\longleftrightarrow` ( $\longleftrightarrow$ ), 97  
`\Longleftrightarrow` ( $\Longleftrightarrow$ ), 97  
`\longmapsto` ( $\longmapsto$ ), 97  
`\longrightarrow` ( $\longrightarrow$ ), 97  
`\Longrightarrow` ( $\Longrightarrow$ ), 97  
`\makebox`, 38, 130  
`\makeglossary`, 153  
`\makeindex`, 151  
`\makelabels`, 185  
`\maketitle`, 74  
`\mapsto` ( $\mapsto$ ), 97  
`\marginpar`, 69  
`\marginparwidth`, 82  
`\marginparsep`, 82  
`\marginparpush`, 82  
`\math-with-secnums`  
 (APS 版式专用命令), 194  
`\mathindent`, 90  
`\max` ( $\max$ ), 100  
`\mbox`, 39, 110  
`\mbox{}`, 109  
`\mediumtext`, 189  
`\medskip`, 38  
`\medskipamount`, 36  
`\mho` ( $\mho$ ), 97  
`\mid` ( $\mid$ ), 96  
`\min` ( $\min$ ), 100  
`\mit` (数学斜体), 27

- $\mathbb{N}$ , 34  
 $\backslash\models$  ( $\models$ ), 96  
 $\backslashmoreauthors$  (APS 版式专用命令), 190  
 $\backslashmp$  ( $\mp$ ), 96  
 $\backslashmu$  ( $\mu$ ), 95  
 $\backslashmulticolumn$ , 121  
 $\backslashmultiput$ , 128, 141  
 $\backslashnabla$  ( $\nabla$ ), 97  
 $\backslashname$ , 184  
 $\backslashnarrowtext$ , 189  
 $\backslashnatural$  ( $\natural$ ), 97  
 $\backslashnearrow$  ( $\nearrow$ ), 97  
 $\backslashneg$  ( $\neg$ ), 97  
 $\backslashneq$  ( $\neq$ ), 96  
 $\backslashnewcommand$ , 155  
 $\backslashnewcounter$ , 171  
 $\backslashnewenvironment$ , 159, 160  
 $\backslashnewfont$ , 177  
 $\backslashnewline$ , 12  
 $\backslashnewpage$ , 13, 73  
 $\backslashnewsavebox$ , 42, 139  
 $\backslashnewtheorem$ , 171, 173  
 $\backslashni$  ( $\ni$ ), 96  
 $\backslashnolinebreak[n]$ , 180  
 $\backslashnonumber$ , 108  
 $\backslashnopagebreak[n]$ , 181  
 $\backslashnormalsize$ , 30  
 $\backslashnot$ , 98  
 $\backslashnu$  ( $\nu$ ), 95  
 $\backslashnwarrow$  ( $\nwarrow$ ), 97  
 $\circ$  ( $\circ$ ), 95  
 $\backslasho$  ( $\leq$ ), 44  
 $\backslashO$  ( $\vee$ ), 44  
 $\backslashOE$  ( $\equiv$ ), 44  
 $\backslashoddsidemargin$ , 81  
 $\backslashodot$  ( $\odot$ ), 96  
 $\backslashoe$  ( $\circ$ ), 44  
 $\backslashoint$  ( $\oint$ ), 98  
 $\backslashomega$  ( $\omega$ ), 95  
 $\backslashOmega$  ( $\Omega$ ), 95  
 $\backslashominus$  ( $\ominus$ ), 96  
 $\backslashonecolumn$ , 79  
 $\backslashopening$ , 185  
 $\backslashoplus$  ( $\oplus$ ), 96  
 $\backslashoslash$  ( $\oslash$ ), 96  
 $\backslashotimes$  ( $\otimes$ ), 96  
 $\backslashoval$ , 137  
 $\backslashover$ , 101  
 $\backslashoverbrace$ , 111  
 $\backslashoverline$ , 110  
 $\backslashowns$ , 96  
 $p$ , 115, 125  
 $p\{\dots\}$ , 123  
 $\backslashP$  ( $\P$ ), 43  
 $\backslashpacs$  (APS 版式专用命令), 191  
 $\backslashpagebreak[n]$ , 181  
 $\backslashpagenumbering$ , 70  
 $\backslashpageref$ , 147  
 $\backslashpagestyle$ , 71  
 $\backslashpar$ , 12  
 $\backslashparagraph$ , 66  
 $\backslashparallel$  ( $\parallel$ ), 96  
 $\backslashparbox$ , 77  
 $\backslashparindent$ , 83  
 $\backslashparsep$ , 174  
 $\backslashparskip$ , 83  
 $\backslashpart$ , 66  
 $\backslashpartial$  ( $\partial$ ), 97  
 $\backslashpartopsep$ , 174  
 $\backslashperp$  ( $\perp$ ), 96  
 $\backslashphi$  ( $\phi$ ), 95  
 $\backslashPhi$  ( $\Phi$ ), 95  
 $\backslashpi$  ( $\pi$ ), 95  
 $\backslashPi$  ( $\Pi$ ), 95  
 $\backslashpm$  ( $\pm$ ), 96, 102  
 $\backslashpmod$ , 102  
 $\backslashpoptabs$ , 118  
 $\backslashpounds$  ( $\pounds$ ), 43  
 $\backslashPr$  ( $\Pr$ ), 100  
 $\backslashprec$  ( $<$ ), 96  
 $\backslashpreceq$  ( $\leq$ ), 96  
 $\backslashprime$  ( $'$ ), 97  
 $\backslashprod$  ( $\prod$ ), 98  
 $\backslashpropto$  ( $\propto$ ), 96  
 $\backslashprotect$ , 219  
 $\backslashps$ , 186

- \psi ( $\psi$ ), 95
- \Psi ( $\Psi$ ), 95
- pt, 34
- \pushtabs, 118
- \put, 128
- r (右齐), 38, 103, 120, 122, 129, 138
- \raggedleft, 57
- \raggedright, 57
- \raisebox, 41
- \rangle ( $\rangle$ ), 106
- \rceil ( $\rceil$ ), 106
- \Re ( $\Re$ ), 97
- \receipt (APS 版式专用命令), 190
- \ref, 145
- \refstepcounter, 171
- \renewcommand, 158
- \renewenvironment, 162
- \rfloor ( $\rfloor$ ), 106
- \rhd ( $\rhd$ ), 96
- \rho ( $\rho$ ), 95
- \right, 106
- \rightarrow ( $\rightarrow$ ), 97
- \rightharpoonupdown ( $\rightharpoonupdown$ ), 97
- \rightharpoonup ( $\rightharpoonup$ ), 97
- \rightleftharpoons ( $\rightleftharpoons$ ), 97
- \rightline, 55
- \rightmargin, 175
- \rm (罗马字体), 27
- \roman, 170
- \Roman (罗马字体), 170
- \rule, 40, 131
- \S ( $\S$ ), 25, 43
- \samepage, 181
- \savebox, 42, 139
- \sbox, 42, 139
- \sc (小体大写字体), 27
- \scriptsize, 30
- \searrow ( $\searrow$ ), 97
- \sec (sec), 100
- \section, 66
- \section\*, 67
- \setcounter, 169
- \setlength, 37, 127
- \setminus ( $\setminus$ ), 96
- \settowidth, 37
- \sf (等线体), 27
- \sharp ( $\sharp$ ), 97
- \shortstack, 135
- \sigma ( $\sigma$ ), 95
- \Sigma ( $\Sigma$ ), 95
- \signature, 184
- \sim ( $\sim$ ), 96
- \simeq ( $\simeq$ ), 96
- \sin (sin), 100
- \sinh (sinh), 100
- \sl (倾斜体), 27
- \sloppy, 180
- \small, 30
- \smallskip, 36
- \smallskipamount, 36
- \smile ( $\smile$ ), 96
- \songti (汉字宋体. CCT 系统专用命令), 216
- \spadesuit ( $\spadesuit$ ), 97
- \special, 219
- \sqcap ( $\sqcap$ ), 96
- \sqcup ( $\sqcup$ ), 96
- \sqrt, 101
- \sqsubset ( $\sqsubset$ ), 96
- \sqsubseteq ( $\sqsubseteq$ ), 96
- \sqsupset ( $\sqsupset$ ), 96
- \sqsupseteq ( $\sqsupseteq$ ), 96
- \ss ( $\ss$ ), 44
- \stackrel, 113
- \star ( $\star$ ), 96
- \stepcounter, 171
- \subitem, 152
- \subparagraph, 66
- \subsection, 66
- \subset ( $\subset$ ), 96
- \subseteq ( $\subseteq$ ), 96
- \subsubsection, 66
- \subsubitem, 152
- \succ ( $\succ$ ), 96
- \succeq ( $\succeq$ ), 96
- \sum ( $\sum$ ), 98

- $\backslash sup$  (sup), 100  
 $\backslash supset$  ( $\supset$ ), 96  
 $\backslash supseteq$  ( $\supseteq$ ), 96  
 $\backslash surd$  ( $\sqrt{\phantom{x}}$ ), 97  
 $\backslash swarrow$  ( $\swarrow$ ), 97  
 $t$  (顶部齐), 76, 105, 116, 125, 129, 138  
 $\backslash t$ , 44  
 $\backslash tableline$  (APS 版式专用命令), 194  
 $\backslash tablenotes$  (APS 版式专用命令), 194  
 $\backslash tableofcontents$ , 73, 116, 144  
 $\backslash tan$  (tan), 100  
 $\backslash tanh$  (tanh), 100  
 $\backslash tau$  ( $\tau$ ), 95  
 $\backslash telephone$ , 184  
 $\backslash TeX$  (TeX), 9  
 $\backslash textheight$ , 37, 79  
 $\backslash textwidth$ , 37, 79  
 $\backslash thanks$ , 74  
 $\backslash the \dots$ , 169, 171  
 $\backslash thechapter$ , 169  
 $\backslash thefootnote$ , 170  
 $\backslash thepage$ , 169  
 $\backslash thesection$ , 169, 170  
 $\backslash thesubsection$ , 171  
 $\backslash theta$  ( $\theta$ ), 95  
 $\backslash Theta$  ( $\Theta$ ), 95  
 $\backslash thicklines$ , 128  
 $\backslash thinlines$ , 128  
 $\backslash thispagestyle$ , 72  
 $\backslash tilde$  ( $\sim$ ), 112  
 $\backslash times$  ( $\times$ ), 96  
 $\backslash tiny$ , 30  
 $\backslash title$ , 74  
 $\backslash today$ , 26, 74  
 $\backslash tolerance$  (APS 版式专用命令), 188  
 $\backslash top$  ( $\top$ ), 97  
 $\backslash topmargin$ , 80  
 $\backslash topsep$ , 174  
 $\backslash triangle$  ( $\triangle$ ), 97  
 $\backslash triangleleft$  ( $\triangleleft$ ), 96  
 $\backslash triangleleftarrow$  ( $\triangleleft$ ), 96  
 $\backslash tt$  (打字机字体), 27  
 $\backslash twocolumn$ , 79  
 $\backslash typein$ , 163  
 $\backslash typeout$ , 162  
 $\backslash u$ , 44  
 $\backslash underbrace$ , 111  
 $\backslash underline$ , 110  
 $\backslash unitlength$ , 127  
 $\backslash unlhd$  ( $\leq$ ), 96  
 $\backslash unrhd$  ( $\geq$ ), 96  
 $\backslash uparrow$  ( $\uparrow$ ), 97, 106  
 $\backslash Uparrow$  ( $\Uparrow$ ), 97, 106  
 $\backslash updownarrow$  ( $\updownarrow$ ), 97, 106  
 $\backslash Updownarrow$  ( $\Updownarrow$ ), 97, 106  
 $\backslash uplus$  ( $\oplus$ ), 96  
 $\backslash upsilon$  ( $\upsilon$ ), 95  
 $\backslash Upsilon$  ( $\Upsilon$ ), 95  
 $\backslash usebox$ , 42, 140  
 $\backslash usecounter$ , 175  
 $\backslash v$ , 25, 44  
 $\backslash v \{ \dots \}$ , 25  
 $\backslash varepsilon$  ( $\varepsilon$ ), 95  
 $\backslash varphi$  ( $\varphi$ ), 95  
 $\backslash varpi$  ( $\varpi$ ), 95  
 $\backslash varrho$  ( $\varrho$ ), 95  
 $\backslash varsigma$  ( $\varsigma$ ), 95  
 $\backslash vartheta$  ( $\vartheta$ ), 95  
 $\backslash vdash$  ( $\vdash$ ), 96  
 $\backslash vdots$  ( $\vdots$ ), 99  
 $\backslash vec$ , 112  
 $\backslash vector$ , 134  
 $\backslash vee$  ( $\vee$ ), 96  
 $\backslash verb$ , 58  
 $\backslash verb*$ , 12, 59  
 $\backslash vert$  ( $\|$ ), 106  
 $\backslash Vert$  ( $\|$ ), 97, 106  
 $\backslash vfill$ , 85  
 $\backslash vspace$ , 35  
 $\backslash vspace*$ , 36  
 $\backslash wedge$  ( $\wedge$ ), 96  
 $\backslash widehat$ , 112  
 $\backslash widetext$  (APS 版式专用命令), 189  
 $\backslash widetilde$ , 112  
 $\backslash wp$  ( $\wp$ ), 97

\wr (i), 96

\xi (ξ), 95

\Xi (Ξ), 95

\zeta (ζ), 95

\zihao (汉字字号, CCT 系统专用命令),  
217

\ziju (汉字字距, CCT 系统专用命令),  
218

# 后 记

1989年9月,作者应邀到意大利国际理论物理中心(ICTP)作短期访问。在ICTP,作者第一次接触到了 $\text{\LaTeX}$ 。从那以后,就一直被其排版程序语言的易读性以及精美的排版效果所迷住。在过去的4年里,只要是撰写论文,作者就直接在计算机上利用 $\text{\LaTeX}$ 语言来编写,并用软盘往国内外杂志出版社送交论文。

$\text{\LaTeX}$ 在科研、教学和行政管理各方面对作者的帮助实在是太大了。

本书是作者在教学与科研的工作间隙里利用微型计算机断断续续“编写”而成的。从第一次“落笔”到最后一页“稿纸”修改完毕,时间跨度为1年。作者之所以“撰写”此书,主要是为了以下几个目的:

- ① 为读者提供一本使用电脑来“写作”的重要工具书;
- ② 为本科生和研究生提供一本实用教材;
- ③ 向读者证明:一个非出版专业的工作人员也可以利用微型计算机编排出完美的符合国际规范的印刷精品;
- ④ 为我国的出版业早日实现完全计算机化,缩短出版周期出点微薄之力。

本书“编写”和出版过程中的部分工作分别得到了国家863高技术科研基金、同济大学重点课程基金、江苏省靖江市暖通器件二厂的资助;华东理工大学计算机专家居德华教授在百忙之中仔细审阅了此稿,并为本书作序;同济大学出版社总编洪建华、编辑陆菊英和责任编辑莫惠林给予本书的出版提供了热情的帮助;北京大学朱允伦副教授和复旦大学陈灏副教授为本书的撰稿提供了可贵的资料和极有价值的建议;博士生刘念华和本科生刘国安协助完成了部分章节的校对工作;硕士生曹阳帮助完成了调用SPT图形的CCT系统辅助接口软件的设计工作。作者谨在此一并表示诚挚的谢意!

最后我要感谢我的妻子陈建华,没有她的鼎力支持,本书的完成是不可能的。

作 者

1993年8月



[ G e n e r a l   I n f o r m a t i o n ]

书名 = L A T E X   智能化科技排版系统

作者 =

页数 = 2 5 8

S S 号 = 0

出版日期 =

封面页  
书名页  
版权页  
前言页  
目录页  
序

## 1 系统简介

### 1.1 引言

### 1.2 系统构成

#### 1.2.1 L A T E X排版系统

#### 1.2.2 软件配置

#### 1.2.3 西文字库

### 1.3 L A T E X系统的主要排版功能

### 1.4 背景知识和必要条件

## 2 L A T E X使用入门

### 2.1 基本操作步骤

#### 2.1.1 编辑 L A T E X源文件

#### 2.1.2 执行排版

#### 2.1.3 显示校样

#### 2.1.4 打印输出

### 2.2 小样文件

#### 2.2.1 文件版式命令

#### 2.2.2 文稿排版环境起始命令

#### 2.2.3 文稿排版环境结束命令

### 2.3 打字与排版入门

#### 2.3.1 空格

#### 2.3.2 换行

#### 2.3.3 分段

#### 2.3.4 换页

#### 2.3.5 拆字

#### 2.3.6 连字

#### 2.3.7 标点符号

#### 2.3.8 控制字符

#### 2.3.9 源程序注释控制符

### 2.4 执行排版

### 2.5 排版输出

#### 2.5.1 排版样式的屏幕显示

#### 2.5.2 宽行针式打印输出

#### 2.5.3 激光打印机输出

## 3 基本行排命令

### 3.1 转码控制符

#### 3.1.1 控制字符类命令

#### 3.1.2 控制词类命令

### 3.2 字体转换命令

### 3.3 字号转换命令

### 3.4 分组控制符 - - 花括号用途之一

### 3.5 参变量控制符 - - 花括号用途之二

### 3.6 尺寸控制命令

#### 3.6.1 L A T E X长度计量单位

#### 3.6.2 L A T E X横向空白间距控制命令

#### 3.6.3 L A T E X纵向空白间距控制命令

#### 3.6.4 长度控制命令的赋值命令

### 3.7 L R排版盒子

#### 3.7.1 无框线 L R排版盒子

#### 3.7.2 带框线 L R排版盒子

#### 3.7.3 实心矩形盒子

#### 3.7.4 L R排版盒子的版面控制

### 3.8 无框线 L R排版盒子的存取

#### 3.8.1 存贮库的命名

#### 3.8.2 无框线 L R排版盒子的存贮

- 3 . 8 . 3 存贮信息的调用
  - 3 . 9 西文特殊符号的排版控制命令
  - 3 . 1 0 L A T E X 约定
- 4 排版环境命令
  - 4 . 1 独立排版环境的进入与退出
  - 4 . 2 引文环境命令
    - 4 . 2 . 1 命令格式 1
    - 4 . 2 . 2 命令格式 2
  - 4 . 3 诗句环境命令
  - 4 . 4 行居中环境命令
    - 4 . 4 . 1 多行居中命令
    - 4 . 4 . 2 单行居中命令
  - 4 . 5 行右齐和行左齐环境命令
    - 4 . 5 . 1 行右齐命令
    - 4 . 5 . 2 行左齐命令
  - 4 . 6 引文环境中的行对齐命令
  - 4 . 7 抄录环境命令
    - 4 . 7 . 1 多行抄录环境命令
    - 4 . 7 . 2 单行抄录命令
  - 4 . 8 条目列表环境命令
    - 4 . 8 . 1 编号条目列表环境
    - 4 . 8 . 2 无序号条目列表环境
    - 4 . 8 . 3 描述型编码条目列表环境
- 5 版式处理命令
  - 5 . 1 章节排版命令
  - 5 . 2 脚注命令
  - 5 . 3 旁注命令
  - 5 . 4 页码和书眉排版命令
    - 5 . 4 . 1 页码数字字体命令
    - 5 . 4 . 2 页眉版式命令
  - 5 . 5 附录版式命令
  - 5 . 6 列目录命令
  - 5 . 7 题名排版命令
  - 5 . 8 小页排版命令
    - 5 . 8 . 1 小页排版环境命令
    - 5 . 8 . 2 小页排版盒子命令
  - 5 . 9 摘要排版控制命令
  - 5 . 1 0 分栏排版控制命令
  - 5 . 1 1 版式全局说明控制命令
    - 5 . 1 1 . 1 版心尺寸控制命令
    - 5 . 1 1 . 2 页边距控制命令
    - 5 . 1 1 . 3 页眉尺寸控制命令
    - 5 . 1 1 . 4 脚注尺寸控制命令
    - 5 . 1 1 . 5 旁注尺寸控制命令
    - 5 . 1 1 . 6 行距控制命令
    - 5 . 1 1 . 7 段落间距控制命令
    - 5 . 1 1 . 8 段首缩排控制命令
    - 5 . 1 1 . 9 栏间空距排版命令
  - 5 . 1 2 版面撑满控制命令
    - 5 . 1 2 . 1 横向撑满控制命令
    - 5 . 1 2 . 2 纵向撑满控制命令
    - 5 . 1 2 . 3 点线撑满控制命令
- 6 数学公式排版命令
  - 6 . 1 数学排版概述
    - 6 . 1 . 1 行中数学公式状态命令
    - 6 . 1 . 2 独立数学公式状态命令
    - 6 . 1 . 3 独立编号数学方程命令
  - 6 . 2 数学公式状态下的基本规则
    - 6 . 2 . 1 字距
    - 6 . 2 . 1 字体

- 6 . 2 . 3 数学符号
- 6 . 2 . 4 公式的分行和行距
- 6 . 3 上下标命令
  - 6 . 3 . 1 上标排版命令
  - 6 . 3 . 2 下标排版命令
- 6 . 4 数学符号排版命令
  - 6 . 4 . 1 希腊字母
  - 6 . 4 . 2 书写体
  - 6 . 4 . 3 特殊数学符号
  - 6 . 4 . 4 可变型数学符号
  - 6 . 4 . 5 数学公式中的省略号
- 6 . 5 对数型函数排版命令
- 6 . 6 分式、根式和取模的排版命令
  - 6 . 6 . 1 分式
  - 6 . 6 . 2 根式
  - 6 . 6 . 3 取模
- 6 . 7 矩阵排版命令
  - 6 . 7 . 1 矩阵元排版环境命令
  - 6 . 7 . 2 矩阵的对位参量
- 6 . 8 界标排版命令
  - 6 . 8 . 1 静态配置界标
  - 6 . 8 . 2 动态配置界标
- 6 . 9 多行数学方程式排版命令
  - 6 . 9 . 1 多行编号方程式排版命令
  - 6 . 9 . 2 多行无编号方程式排版命令
- 6 . 1 0 数学符号的修饰
  - 6 . 1 0 . 1 划横线命令
  - 6 . 1 0 . 2 卧式花括号命令
  - 6 . 1 0 . 3 “戴帽”命令
  - 6 . 1 0 . 4 “堆砌”命令

## 7 表格排版命令

- 7 . 1 活动表格
  - 7 . 1 . 1 表格排版环境命令
  - 7 . 1 . 2 表格标题命令
  - 7 . 1 . 3 表格目录命令
- 7 . 2 无框线表格制表命令
  - 7 . 2 . 1 无框线表格排版环境命令
  - 7 . 2 . 2 表文排版控制命令
- 7 . 3 可划线表格制表命令
  - 7 . 3 . 1 可划线表格排版环境命令
  - 7 . 3 . 2 表格划线命令
- 7 . 4 对位表达式
  - 7 . 4 . 1 ? - 表达式
  - 7 . 4 . 2 p - 表达式
  - 7 . 4 . 3 ? - 表达式

## 8 L A T E X 绘图命令

- 8 . 1 插图环境命令
  - 8 . 1 . 1 插图标题命令
  - 8 . 1 . 2 插图目录命令
- 8 . 2 绘图环境命令
- 8 . 3 基本作图命令
  - 8 . 3 . 1 “粘贴”文字
  - 8 . 3 . 2 画矩形
  - 8 . 3 . 3 画直线
  - 8 . 3 . 4 画箭头
  - 8 . 3 . 5 堆积字符串
  - 8 . 3 . 6 画圆和圆弧
- 8 . 4 无线矩形图形的存取
  - 8 . 4 . 1 图形存贮库的命名
  - 8 . 4 . 2 无线矩形图形的存贮

- 8.4.3 存贮图形的调用
  - 8.5 规则平移场合下图形的复制
- 9 交叉引用
  - 9.1 章节图表目录
    - 9.1.1 章节目录的生成
    - 9.1.2 图表目录的生成
  - 9.2 章节图表的交叉引用
    - 9.2.1 引用标识符
    - 9.2.2 章节的交叉引用
    - 9.2.3 方程的交叉引用
    - 9.2.4 图表的交叉引用
    - 9.2.5 页面的交叉引用
  - 9.3 参考文献
    - 9.3.1 参考文献的编排
    - 9.3.2 参考文献的引用
  - 9.4 编制索引
    - 9.4.1 生成索引信息文件
    - 9.4.2 编制文件索引
  - 9.5 编制词汇表
- 10 L A T E X 程序设计高级技巧
  - 10.1 自行定义 L A T E X 控制命令
    - 10.1.1 定义一个排版子过程控制命令
    - 10.1.2 含变量控制命令的自定义方法
    - 10.1.3 重新定义已经设置的控制命令
  - 10.2 自行定义 L A T E X 排版环境
    - 10.2.1 排版环境控制命令的化简
    - 10.2.2 自行定义排版环境控制命令
    - 10.2.3 含变量排版环境的自定义方法
    - 10.2.4 重新定义已经设置的排版环境
  - 10.3 人机对话
    - 10.3.1 终端显示命令
    - 10.3.2 终端输入命令
  - 10.4 调入另外一个 L A T E X 源文件
    - 10.4.1 无选择方式调入 L A T E X 子文件
    - 10.4.2 有选择调入 L A T E X 子文件
  - 10.5 如何改变系统的计数方式
    - 10.5.1 计数器和计数方式
    - 10.5.2 为计数器赋值
    - 10.5.3 计数器数字打印命令
    - 10.5.4 改变序码的字体形式
  - 10.6 自定义排版计数器
  - 10.7 定理排版环境命令
  - 10.8 列表环境 ( l i s t ) 的构造
  - 10.9 如何使用新的字体
  - 10.10 断行控制命令
    - 10.10.1 断字注解命令
    - 10.10.2 断行控制命令
    - 10.10.3 自然段断字处理环境命令
  - 10.11 断页控制命令
- 11 信件类 ( l e t t e r ) 版式和 A P S 版式
  - 11.1 信件类版式
    - 11.1.1 信件类 ( l e t t e r ) 文件版式命令
    - 11.1.2 发信人有关通讯信息
    - 11.1.3 信件排版环境
    - 11.1.4 信件正文的引出和结尾
    - 11.1.5 信件附加信息
    - 11.1.6 信件排版实例
  - 11.2 A P S 科技论文版式
    - 11.2.1 A P S 宏程序库文件的结构和安装
    - 11.2.2 A P S 文件版式的环境设置

	1 1 . 2 . 3	A P S 版式的尺寸控制
	1 1 . 2 . 4	题名版面
	1 1 . 2 . 5	节次标题
	1 1 . 2 . 6	脚注和参考文献
	1 1 . 2 . 7	插图与表格版面的设计
	1 1 . 2 . 8	数学公式的编排
1 2	源程序语法错误的发现和处理	
	1 2 . 1	出错定位
	1 2 . 2	T E X 出错信息
	1 2 . 3	L A T E X 出错信息
	1 2 . 4	语法严重错误
	1 2 . 4 . 1	无文稿排版环境
	1 2 . 4 . 2	无文件版式命令
	1 2 . 4 . 3	无效文件版式
	1 2 . 4 . 4	数学状态中的意外语法错误
	1 2 . 5	T E X 系统警告信息
	1 2 . 5 . 1	断行警告信息
	1 2 . 5 . 2	断页警告信息
	1 2 . 6	L A T E X 系统警告信息
附录 A	英文字体	
附录 B	C C T 中西文排版系统简介	
	B . 1	C C T 系统的主要功能和特点
	B . 2	C C T 中西文排版系统的构成
	B . 3	C L A T E X 系统的基本流程
	B . 4	C C T 初始化程序 C C T I N I T . E X E
	B . 5	C C T 预处理程序 C C T . E X E
	B . 6	字号定义文件 C C T . D A T
	B . 7	C L A T E X 专用排版命令
	B . 7 . 1	C L A T E X 源文件的版式
	B . 7 . 2	汉字字体转换命令
	B . 7 . 3	汉字字号转换命令
	B . 7 . 4	汉字尺寸控制命令
	B . 7 . 5	C L A T E X 与西文 L A T E X 的兼容性
	B . 8	图形接口
	B . 8 . 1	图形拼版命令
	B . 8 . 2	B M F 图形文件的生成及数据格式
附录 C	作图辅助软件 T E X c a d 简介	
	C . 1	引言
	C . 2	T E X c a d 软件运行环境
	C . 3	T E X c a d 的启动和退出
	C . 4	菜单选择
	C . 5	信息窗口
	C . 6	使用 T E X c a d 窗口绘制图形
	C . 6 . 1	“ 粘贴 ” 文字
	C . 6 . 2	实线、虚线和实心矩形盒子
	C . 6 . 3	直线与箭头
	C . 6 . 4	圆和圆角矩形
	C . 7	图像编辑
	C . 7 . 1	图形定义
	C . 7 . 2	文字编辑
	C . 7 . 3	图像复制和移动
	C . 7 . 4	图像删除
	C . 7 . 5	图形或画面的剪取和拼嵌
	C . 8	其它功能
	C . 9	绘图源程序的调入和生成
附录 D	出错信息	
	D . 1	L A T E X 系统出错信息
	D . 2	T E X 系统出错信息
附录 E	常用 L A T E X 命令	
参考文献		

## 索引

## 后记

- 3 . 1 西文字体转换命令表
- 3 . 2 字号转换命令表
- 3 . 3 不同基准字体状态下字号转换命令控制的实际尺寸
- 3 . 4 各种字号的有效字体详表
- 3 . 5 常用的 L A T E X 长度计量单位表
- 3 . 6 特殊标识符号命令表
- 3 . 7 西文 ( 非英语类 ) 特殊符号命令表
- 3 . 8 字母外形特征符号修饰控制命令表
- 5 . 1 章节排版命令表
- 5 . 2 页码形式参数表
- 6 . 1 数学公式状态字距命令表
- 6 . 2 希腊字母控制命令表
- 6 . 3 数学运算符表
- 6 . 4 确定型逻辑运算符表
- 6 . 5 箭头符号表
- 6 . 6 花样符号表
- 6 . 7 否定式逻辑运算符表
- 6 . 8 可变型数学符号表
- 6 . 9 省略号控制命令表
- 6 . 1 0 对数型函数排版命令表
- 6 . 1 1 矩阵排版环境命令列参数表
- 6 . 1 2 矩阵元排版命令对位参数表
- 6 . 1 3 常用界标排版命令表
- 6 . 1 4 数学符号 “ 戴帽 ” 命令表
- 1 0 . 1 L A T E X 排版计数器标识名表
- 1 0 . 2 序码数字字体控制命令表
- 1 0 . 3 常用排版字库文件引用名表
- B . 1 汉字字体转换命令表
- 2 . 1 L A T E X 排版系统流程图
- 2 . 2 显示系统的选择菜单
- 2 . 3 激光打印驱动程序 ( P T I H P . E X E ) 屏幕显示菜单
- 5 . 1 A 4 纸版式全局说明示意图
- 8 . 1 画实线矩形示意图
- 8 . 2 虚线矩形作图实例
- 8 . 3 画直线示意图
- 8 . 4 矢量 ( 箭头 ) 绘制示意图
- 8 . 5 空心圆和实心圆绘制示意图
- 8 . 6 圆角矩形绘图方式示意图
- 8 . 7 箭头或直线与圆弧光滑连接示意图
- 8 . 8 无线矩形图形的存取示意图
- 8 . 9 箭头线性平移复制示意图
- 8 . 1 0 实心圆两维复制示意图
- 1 0 . 1 提法信息终端显示示意图
- 1 0 . 2 文字信息终端输入示意图
- 1 0 . 3 为自定义控制命令赋字符串的屏幕显示示意图
- 1 0 . 4 条目列表环境 ( l i s t ) 版面控制示意图
- C . 1 T E X c a d 作图窗口示意图
- C . 2 T E X c a d 主菜单和子菜单结构图

## 附录页